GEFORCE NOW SDK 1.11

INTEGRATOR PRIMER | March 2023

The GeForce NOW SDK is a set of APIs, components, and samples that enable developers to seamlessly add the power of cloud gaming to their ecosystem.

NVIDIA.

THE POWER OF GEFORCE NOW

APIS AND BENEFITS

APPLICATION

LAUNCHER / STORE	GAME	ANTI-CHEAT

SDK APIs

STREAM LAUNCH	ACCOUNT LINKING & SINGLE SIGN-ON	ENVIRONMENT CHECK	DEEP LINKING	CLIENT & SESSION INFO

GEFORCE NOW

Single Sign-On	Play Without Installing	Launch In Seconds	Premium GFX Anywhere	Always Up-To-Date	Robust Gaming

NVIDIA.

TOPICS COVERED
BY TARGET AUDIENCE

This document provides a high-level overview of user and developer benefits along with integration flows, interfaces and processes required to stream.

LAUNCHER INTEGRATION	GAME INTEGRATION	ACCOUNT LINKING	GFN DEEP LINKING
This topic covers the details needed by User Experience Designers and Developers alike to integrate the GeForce NOW SDK into an existing launcher application, as well as obtain information about client systems and streaming settings while running inside GeForce NOW. An example launcher application is shown along with source provided in the SDK.	This topic covers the details on APIs available to games running in GeForce NOW to obtain various client system information and streaming session state. This topic is for Game Developers and Product Managers that want their games to run seamlessly in GeForce NOW.	This topic covers linking third-party accounts with a user's GeForce NOW account to implement Single Sign-On to provide a seamless integration experience. Understanding Identity Management is key for Developers.	This topic covers linking into the GeForce NOW web browser client via URL deep links. This topic is for Game Publishers that wish to publish links to their games in GFN as well as Gaming Platforms that wish to link users to playing games in GFN. Example URLs are shown along with documentation provided in the SDK.

LAUNCHER INTEGRATION

ADDING CLOUD GAMING

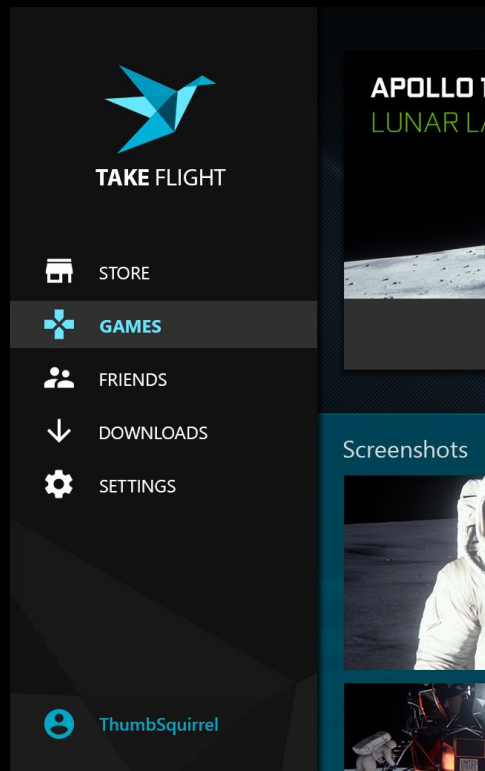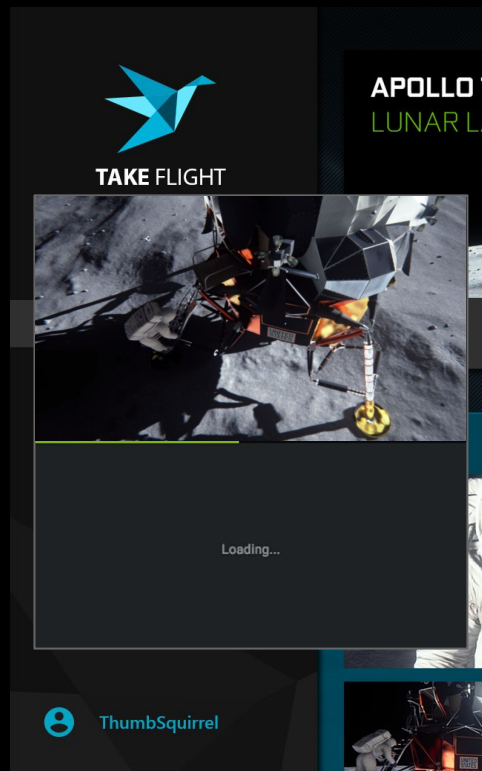# STREAM GAMES DIRECTLY FROM YOUR LAUNCHER
EXAMPLE EXPERIENCE

**Single Sign-On
w/ Linked Accounts**

**Seamless
Integration**

**Launch
In Seconds**

**Premium Experience
Everywhere**

ECOSYSTEM

COMPONENTS AND CONNECTIONS

Local System

PUBLISHER LAUNCHER APP

- Login UI
- Account UI
- Game Launch UI

GeForce NOW SDK Runtime Dynamic Library

- GeForce NOW Client Installation Handler
- SDK Proxy

GeForce NOW Client

- Streamer Client
- SDK Services

CLOUD

NVIDIA IDM

- OpenID Connect Services

NVIDIA Dev Service Console

- Game Metadata

NVIDIA Account Linking Service

- Account Information Database

GeForce NOW Server

- Game Seat

Publisher IDM

- Authentication
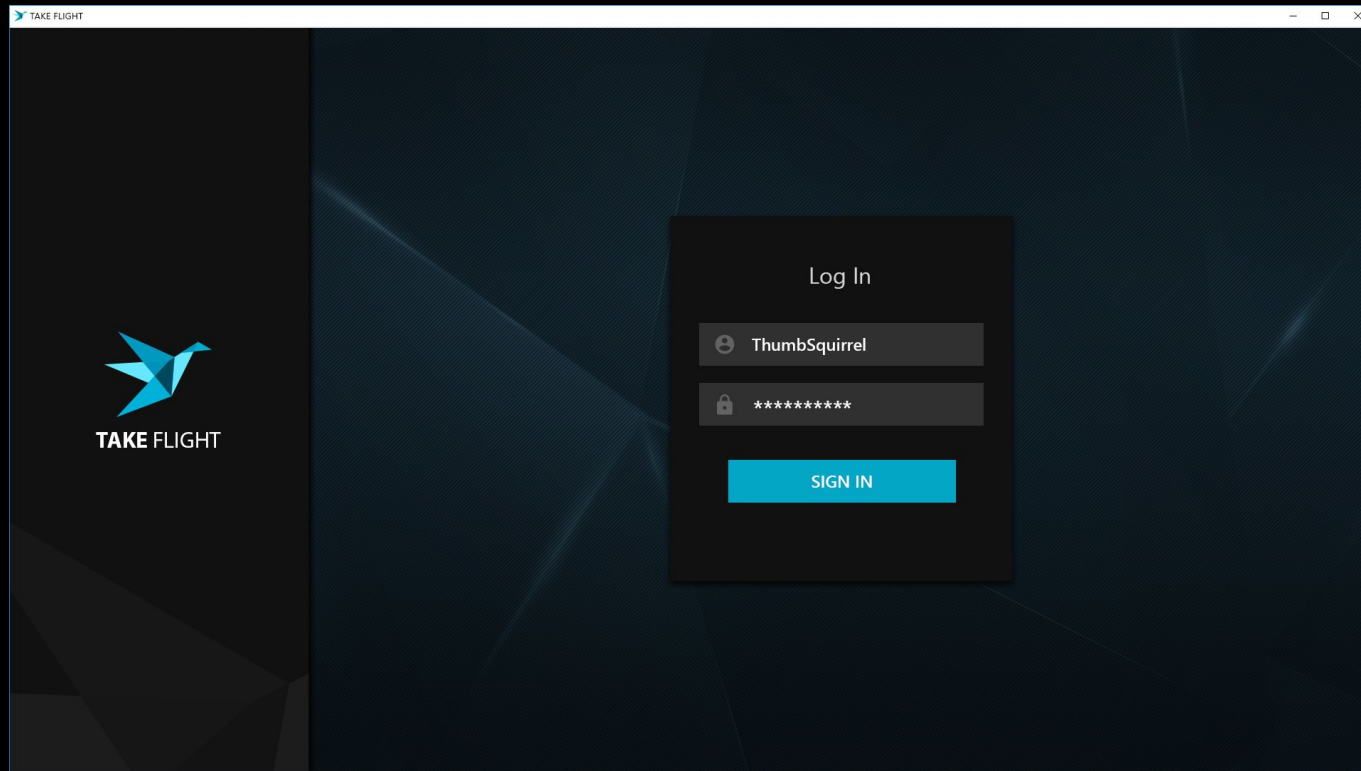
NVIDIA

EXAMPLE LAUNCHER
USER AUTHENTICATION

LOGIN DIALOG

Sign on to the system is handled by the launcher application itself which gives the gamer access to the ecosystem provided by the publisher.

Signing on to the system automatically restores connections to any third-party linked accounts, including the gamer's NVIDIA GeForce NOW account.

Gamers link third-party accounts by opening the Settings page.

TAKE FLIGHT

TAKE FLIGHT

Log In

ThumbSquirrel

****

SIGN IN

NVIDIA.

EXAMPLE LAUNCHER
STREAMING WITH NVIDIA GEFORCE NOW

GAMES PAGE

Gamers are shown an element in the application which opens the streamer window for those games which are supported by NVIDIA GeForce NOW.

The application uses SDK API calls to verify the application is not running inside GFN and to verify a specific game is able to be streamed from GFN.

The application shows the STREAM NOW button, or doesn't, based on whether or not the specific game can be streamed.

NVIDIA.
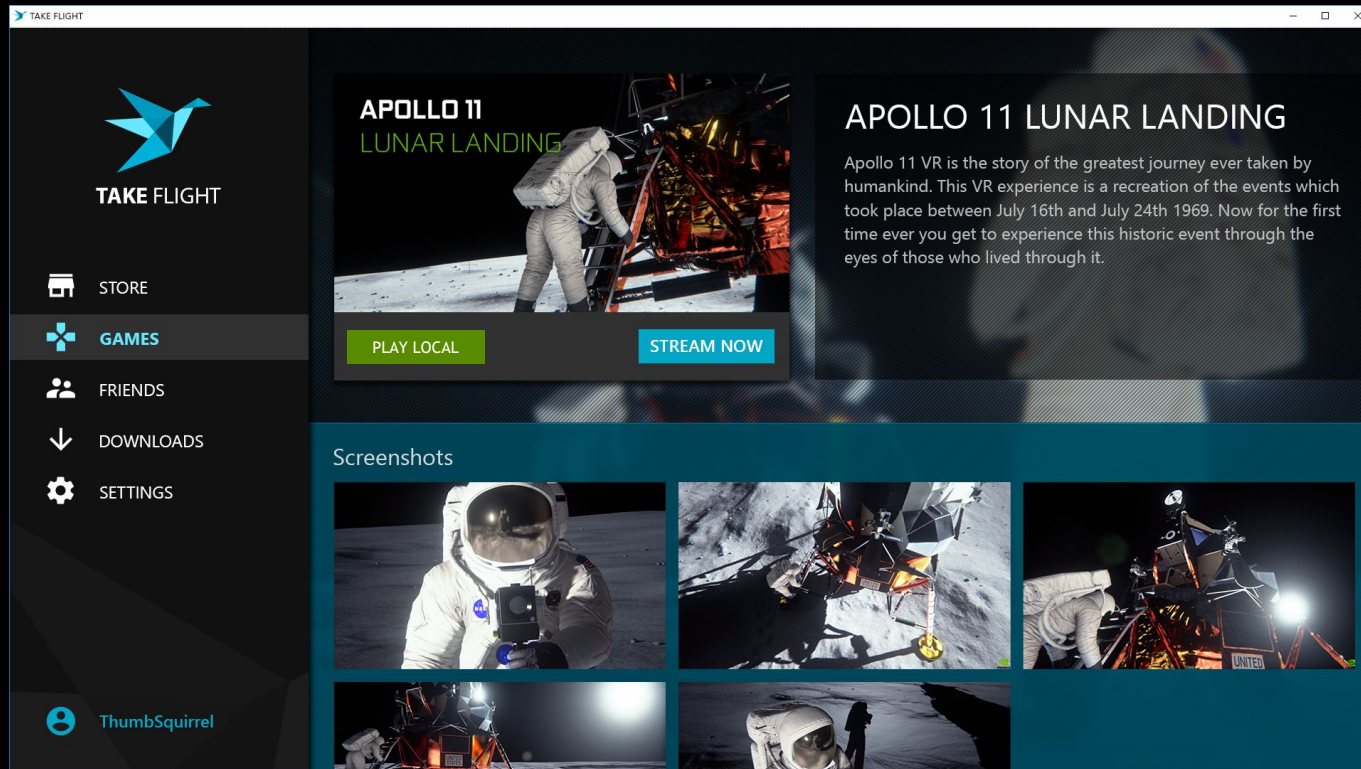
REQUIRED INTEGRATION
ADDING A CLOUD STREAMING OPTION

LOGIC FLOWS

- Check with SDK to confirm not in GFN to know when to allow streaming
- Use SDK to start streaming when user activates UI trigger, which first installs the GeForce NOW client as needed
- Use SDK session callback to know the status of the stream

RELEVANT SDK METHODS

```
GfnRuntimeError gfnInitializeRuntimeSDK(GfnDisplayLanguage);
bool gfnIsRunningInCloud(void);
void gfnStartStreamAsync(StartStreamInput * input, StartStreamCallbackSig
    cb, void * context, unsigned int timeoutMs);
void gfnShutdownRuntimeSDK(void);
```

NVIDIA.

OPTIONAL INTEGRATION
USE OF API WRAPPER FUNCTIONS

Use the C-based API wrapper functions to avoid needing to managing the lifecycle of the SDK library as well as calling export functions.

Use the C-based GfnSdk_SecureLoadLibrary API included with the SDK to automatically validate the SDK libraries to avoid tampering and spoofing.

LOGIC FLOWS

- Load GFN SDK library and initialize the SDK
- Start streaming when user activates UI trigger
- Use SDK callbacks to know the status of the stream
- Release SDK and unload GFN SDK library

RELEVANT SDK METHODS

```
GfnRuntimeError GfnInitializeSDK(gfnDisplayLanguage);

GfnRuntimeError GfnIsRunningInCloud(bool * isRunningInCloud);

GfnRuntimeError GfnIsRunningInCloudSecure(GfnIsRunningInCloudAssurance*);

GfnRuntimeError GfnStartStreamAsync(StartStreamInput * input,
StartStreamCallbackSig cb, void * context, unsigned int timeoutMs);

void gfnShutdownRuntimeSDK();
```

NVIDIA.

SECURE CLOUD ENVIRONMENT DETECTION

USE OF SECURE CLOUD CHECK API

This API checks if running in GeForce NOW game seats in a highly secure fashion.

Useful to decide if high-value features can be enabled or disabled.

The Sample Launcher source include with the SDK provides example code on calling this API.

Starting with SDK 1.10, the calling process *no longer requires elevation*, instead, the binary is added to an NVIDIA API allow list. Refer to the Cloud Check API document in the ./doc folder on how the binary can be added to the allow list.

LOGIC FLOW

- Launch SDK-enabled process with calling binary approved to make the API call
  - `GfnRuntimeError GfnInitializeSDK(gfnDisplayLanguage);`
  - `GfnRuntimeError GfnIsRunningInCloudSecure(GfnIsRunningInCloudAssurance*);`
  - `void gfnShutdownRuntimeSDK();`
- Check returned value for the level of assurance to be running in GFN environment

ASSURANCE VALUES

- gfnNotCloud = Not running in GFN cloud, running local client
- gfnIsCloudLowAssurance = Software heuristics used to determine GFN
- gfnIsCloudMidAssurance = Software and network heuristics to determine GFN
- gfnIsCloudHighAssurance = Hardware heuristics used to determine GFN

GAME INTEGRATION
APIS FOR SEAMLESS INTEGRATION

CLOUD APIS
OBTAINING INFORMATION ABOUT CLIENTS OR GFN SESSION STATE

LOGIC FLOWS

- Check with SDK to know when running in GeForce NOW environment
- Use SDK to obtain various information about the connected client, network performance and streaming session
- Use SDK callbacks to know when GeForce NOW session status or client system changes to update game behavior

RELEVANT SDK METHODS

```
GfnRuntimeError gfnInitializeRuntimeSDK(GfnDisplayLanguage);
bool gfnIsRunningInCloud(void) or GfnRuntimeError GfnIsRunningInCloudSecure(GfnIsRunningInCloudAssurance*);
GfnRuntimeError gfnGetClientInfo(GfnClientInfo* clientInfo);
GfnRuntimeError gfnGetSessionInfo(GfnSessionInfo* sessionInfo);
GfnRuntimeError gfnRegisterClientInfoCallback(ClientInfoCallback clientInfoCallback, void* pUserContext);
GfnRuntimeError gfnRegisterPauseCallback(PauseCallbackSig pauseCallback, void* pUserContext);
GfnRuntimeError gfnRegisterSaveCallback(SaveCallbackSig saveCallback, void* pUserContext);
GfnRuntimeError gfnRegisterExitCallback(ExitCallbackSig exitCallback, void* pUserContext);
void gfnShutdownRuntimeSDK(void);
```

CLOUD APIS
SUPPORT PRE-WARM GAME SESSIONS

With Prewarm game session support, GFN allows users to get into their games faster and skip game loading states and creates a better user experience.

Developers can work with their GFN representatives to integrate this feature into their games.

GAME ONBOARDING FOR PREWARM SESSIONS

- Game integrates GFN SDK to make use of pre-warm APIs
- Game developer works with NVIDIA to onboard prewarm-supported game builds

LOGIC FLOWS

- GFN allocates a game seat for the game launch
- Game loads all generic data (no user specific data as no user is connected)
- Game registers for session initialization callback with SDK and goes idle waiting for callback
- GFN SDK triggers session init callback once a user session is initialized
- Game loads user data and calls gfnAppReady API to notify GFN that it is ready to be displayed
- GFN begins streaming game to user

RELEVANT SDK METHODS

```
GfnRuntimeError gfnRegisterSessionInitCallback(SessionInitCallbackSig
sessionInitCallback, void* pUserContext);
GfnRuntimeError gfnAppReady(bool success, const char * status);
```

ACCOUNT LINKING

FOR SINGLE SIGN-ON

LINKING ACCOUNTS TO SUPPORT SINGLE SIGN-ON
INTEGRATION NECESSARY

The best experience for gamers requires a one time operation to link their store account with an NVIDIA account. Once accomplished, the user enjoys a seamless experience of playing games in GFN without needing to enter their publisher credentials during the streaming session.

ACCOUNT LINKING

- User performs a one time account linking operation from the GFN client.

- Launcher IDM cooperates with NVIDIA Accounting Linking service through OpenID Connect protocols to exchange account information.

- Account information is used to create mapping between two accounts.

SINGLE SIGN-ON

- User links accounts to share account information.

- User streams a game that uses linked accounts.

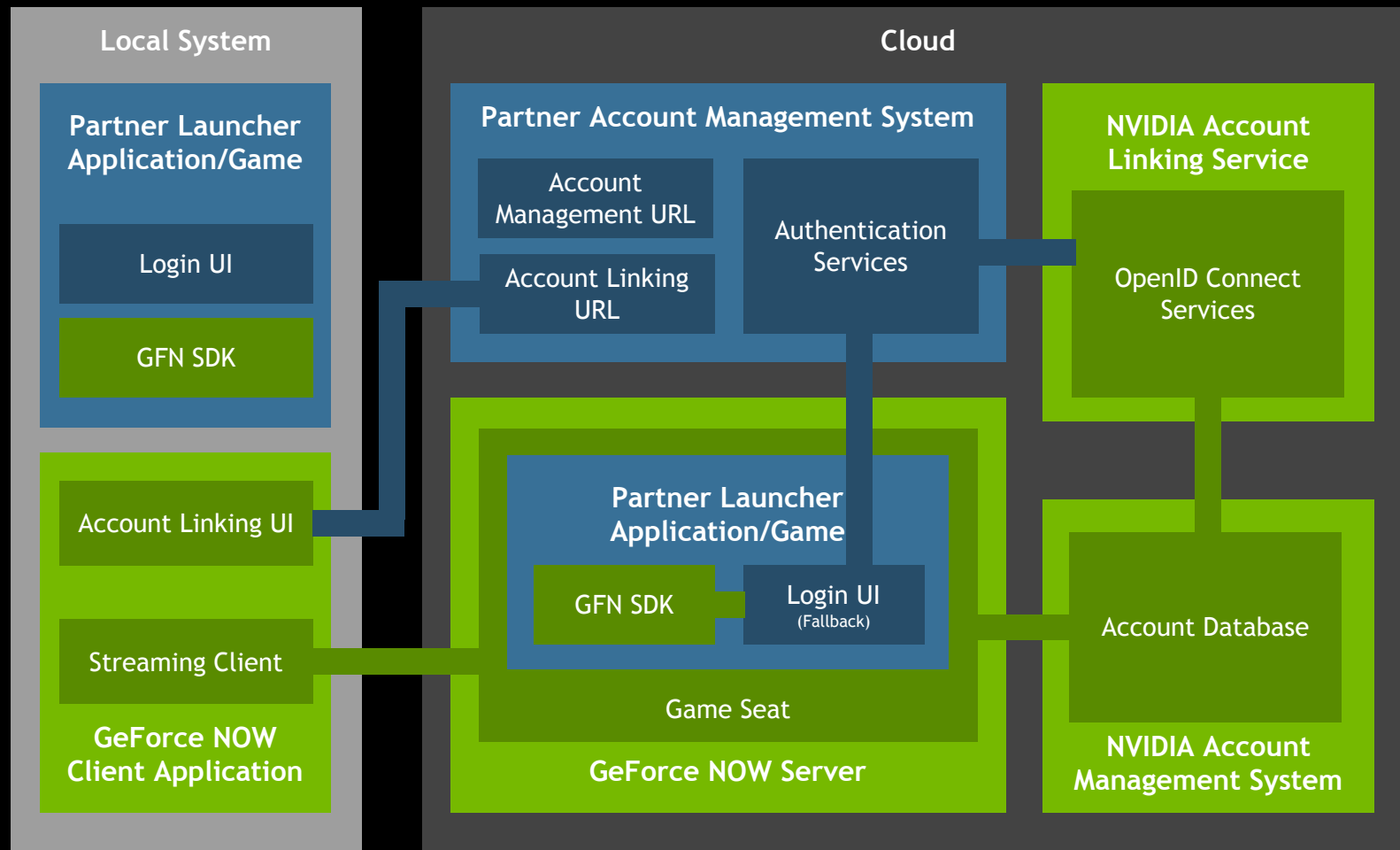- In-stream launcher or game authorizes streaming via account information provided by GFN SDK.

NVIDIA.

ACCOUNT LINKING
COMPONENTS AND CONNECTIONS

GEFORCE NOW ACCOUNT LINKING ARCHITECTURE

This overview illustrates where each component is installed or hosted, and shows important connections between various components involved in account linking and Single Sign-On.

For more detailed information, see the Account Linking and Single Sign-On guide in ./doc folder.

Local System

Partner Launcher Application/Game

Login UI

GFN SDK

Account Linking UI

Streaming Client

GeForce NOW Client Application

Cloud

Partner Account Management System

Account Management URL

Account Linking URL

Authentication Services

NVIDIA Account Linking Service

OpenID Connect Services

Partner Launcher Application/Game

GFN SDK

Login UI (Fallback)

Game Seat

GeForce NOW Server

Account Database

NVIDIA Account Management System

NVIDIA.

EXAMPLE LAUNCHER
THIRD-PARTY ACCOUNT LINKING

SETTINGS PAGE

Gamers who create and link an NVIDIA GeForce NOW account are able to stream games to their device without having to manually enter their publisher credentials.

Linking accounts is accomplished using the industry standard OpenID Connect (OIDC) workflow in a web browser window.

Authorization tokens and user account data are then cached by the backend services so login authorization data can be provided to the publisher application running in the Geforce NOW game seat.

CONNECTIONS
Manage your connected accounts.

Ubisoft
Your account is connected to automatically sign in to games owned on Ubisoft Connect. **DISCONNECT**

Epic Games
Your account is connected to automatically sign in to Fortnite. **DISCONNECT**

Steam
Connect your account to add supported games from your Steam library. **CONNECT**

NVIDIA.

DEEP LINKING

GEFORCE NOW GAME ACCESS VIA URL

INTRODUCING GEFORCE NOW DEEP LINKS

THE PATH TO SHARING CLOUD GAMING

```
https://play.geforcenow.com/games?game-id=<game-id>&utm_source=<utm_source>&utm_campaign=<utm_campaign>
```

"GameWebStore"

"NewLaunch"

A GeForce NOW Deep Link is a specialized URL that directs a user's web browser to the Details page of a specific game in the GFN web browser client.
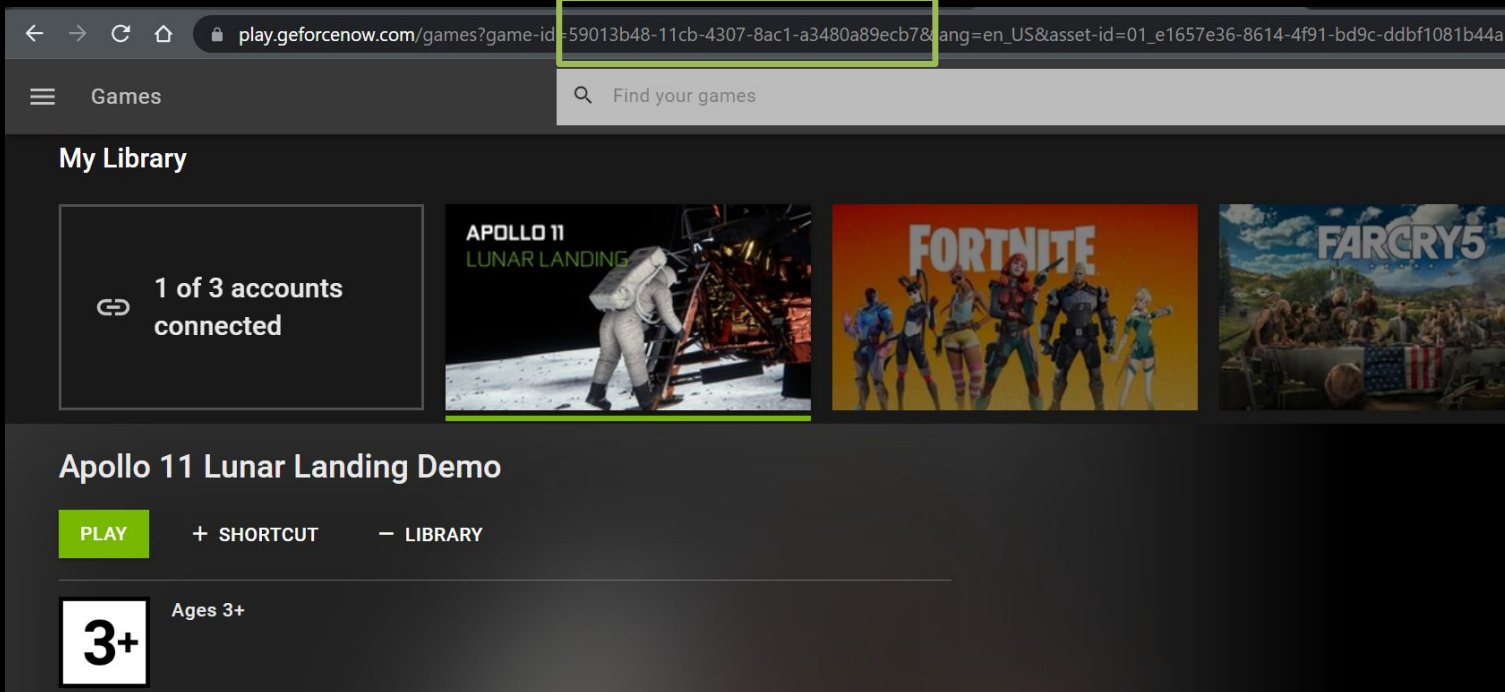
This can be used to invite users via social or web media to discover and play games in GeForce NOW.

The Deep Link requires obtaining the game ID, which can be obtained from the web client's URL in the browser's address bar and a source that is user defined. The source and campaign are user defined.

play.geforcenow.com/games?game-id=59013b48-11cb-4307-8ac1-a3480a89ecb7&lang=en_US&asset-id=01_e1657e36-8614-4f91-bd9c-ddbf1081b44a

Games

🔍 Find your games

My Library

🔗 1 of 3 accounts connected

APOLLO 11 LUNAR LANDING

FORTNITE

FARCRY5

Apollo 11 Lunar Landing Demo

PLAY   + SHORTCUT   — LIBRARY

3+ Ages 3+

NEXT STEPS
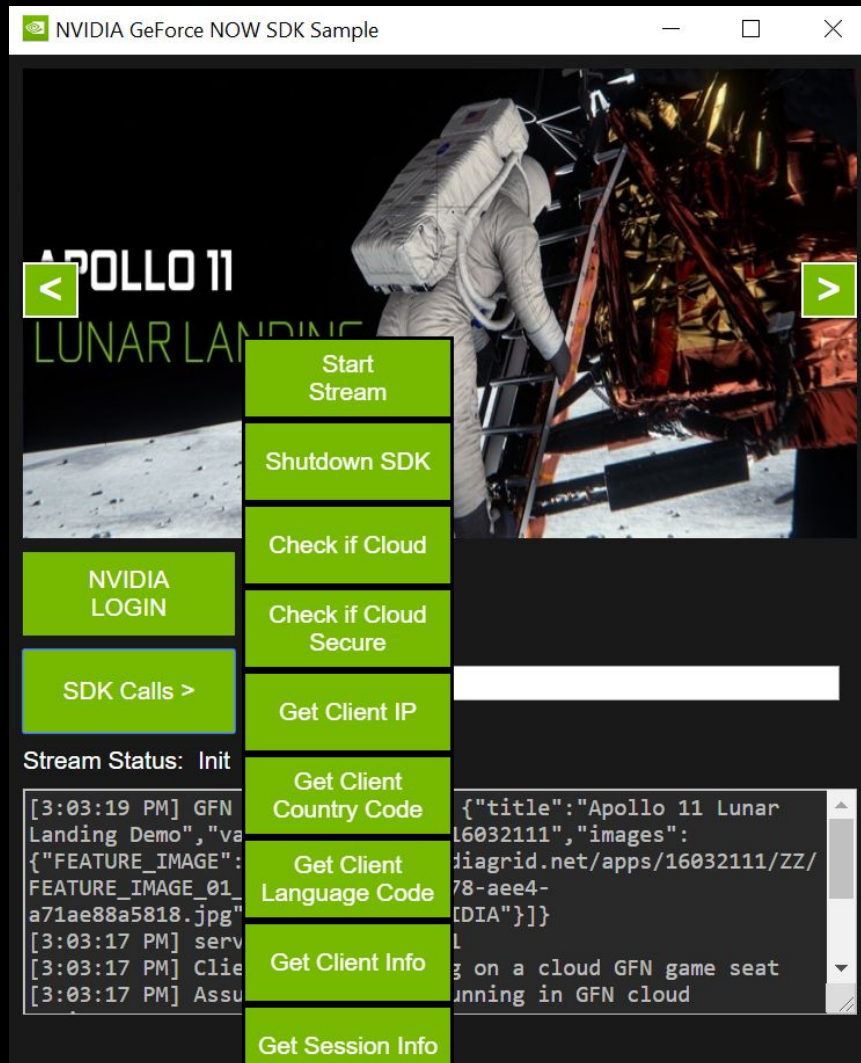INTEGRATING GEFORCE NOW SDK

TRY THE INCLUDED SAMPLES
EXAMPLE CODE FOR ESSENTIAL FLOWS

GEFORCE NOW SDK SAMPLE LAUNCHER APPLICATION

This sample application contains working code with just enough UI controls showing essential integration points and functionality provided by the GeForce NOW SDK.

SDK APIs IN THIS CODE SAMPLE

- Check if running in GeForce NOW cloud environment

- Obtain supported game list

- Start streaming of a game and get stream state status

- Get Client and Session info (when run in Geforce NOW cloud environment)

TRY THE INCLUDED SAMPLES
EXAMPLE CODE FOR ESSENTIAL FLOWS

GEFORCE NOW SDK GAME API SAMPLE

This sample application contains working code for calling game-related APIs available in the SDK when run inside the GFN environment.

SDK APIs IN THIS CODE SAMPLE

- Check if running in GeForce NOW cloud environment

- Registering for Session change callbacks and handling them

- Get Client and Session info

```c
// Example application main
int _tmain(int argc, _TCHAR* argv[])
{
    ApplicationInitialize();

    // Sample C API call
    bool bIsCloudEnvironment = false;
    GfnIsRunningInCloud(&bIsCloudEnvironment);
    printf("\nApplication executing in Geforce NOW environment: %s\n", (bIsCloudEnvironment == true) ? "true" : "false");

    if (bIsCloudEnvironment) // More sample C API calls.
    {
        GfnError runtimeError = gfnSuccess;

        char* clientIp = NULL;
        runtimeError = GfnGetClientIpV4(&clientIp);
        if (runtimeError == gfnSuccess)
        {
            printf("Retrieved Geforce NOW Client I.P.: %s\n", clientIp);
        }
        else
        {
            printf("Failed to retrieve Geforce NOW Client I.P. GfnError: %d\n", (int) runtimeError);
        }

        char* clientLanguageCode = NULL;
        runtimeError = GfnGetClientLanguageCode(&clientLanguageCode);
        if (runtimeError == gfnSuccess)
        {
            printf("Retrieved Geforce NOW client language code: %s\n", clientLanguageCode);
        }
        else
        {
            printf("Failed to retrieve Geforce NOW client language code. GfnError: %d\n", (int) runtimeError);
        }

        char clientCountryCode[3];
        runtimeError = GfnGetClientCountryCode(clientCountryCode, 3);
        if (runtimeError == gfnSuccess)
        {
            printf("Retrieved Geforce NOW client Country code: %s\n", clientCountryCode);
        }
        else
        {
            printf("Failed to retrieve Geforce NOW client Country code. GfnError: %d\n", (int)runtimeError);
```

THANKS FOR USING
THE GEFORCE NOW SDK