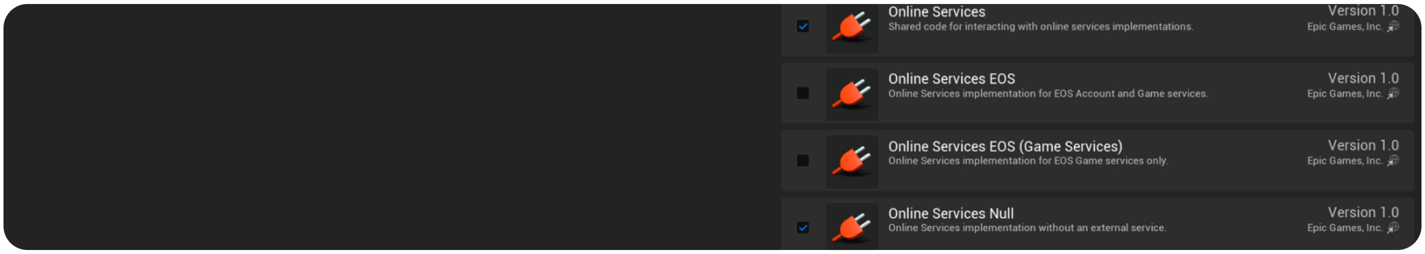
# Setup and Configure the Online Services Plugins

Setup and configure the Online Services plugins for use in your project.



☑ Online Services — Shared code for interacting with online services implementations. — Version 1.0 — Epic Games, Inc.

☐ Online Services EOS — Online Services implementation for EOS Account and Game services. — Version 1.0 — Epic Games, Inc.

☐ Online Services EOS (Game Services) — Online Services implementation for EOS Game services only. — Version 1.0 — Epic Games, Inc.

☑ Online Services Null — Online Services implementation without an external service. — Version 1.0 — Epic Games, Inc.

> ⓘ Learn to use this **Beta** feature, but use caution when shipping with it.

The **Online Services** plugins help you connect various backend online services such as Epic, Steam, Xbox Live, PSN, NLPN, and so on to your **Unreal Engine (UE)** project. This guide shows you how to:

- Enable the Online Services plugins.
- Add the Online Services plugins to your project dependencies.
- Configure the default services for your project.

# Set Up Online Services Plugins

This walkthrough uses the Online Services Null implementation for illustrative purposes. This implementation does not connect to any backend online services and is used for testing purposes. This is a good starting point since the Online Services Null plugin does not require external registration or configuration in order to work with Unreal Engine. For a full list of services supported by the Online Services plugin, see the Online Services Overview documentation.

## Enable the Online Services Plugins

Various Online Services plugins are available to use in your project. The Online Services base plugin is enabled by default.

To enable additional required functionality, follow these steps:

1. Create or open an Unreal Engine C++ Project.
2. From the menu bar, navigate to **Edit > Plugins**. This opens a new window or tab titled **Plugins**.
   a. In this new window, search for "Online Services" or select the **Online Platform** category from the navigation bar on the left-side.
   b. A number of plugins should appear. One of these should be titled **Online Services Null**. Enable the **Online Services Null** plugin by checking the box.
   Enable Online Services Plugins(w:800)

   > If you want to use Epic Online Services for your backend online services, choose **Online Services EOS** instead of **Online Services Null**. Doing so might require you to register your product with Epic Online Services and configure the backend appropriately for the Online Services plugin to operate as expected.

   a. A message appears stating that "You must restart Unreal Editor for changes to take effect." Restart the **Unreal Editor** by clicking **Restart Now**.
3. You have now enabled the Online Services Null plugin in your project.

## Add the Online Services Plugins to Project Dependencies

To use the Online Services plugins in your project's C++ code, you mustadd the plugins to your project module as public dependencies.

To add the plugins to your project module's public dependencies, follow these steps:

1. Open your Unreal Engine C++ Project in the **Unreal Editor**.
2. Open Visual Studio by selecting **Tools > Open Visual Studio**. This opens your project's C++ source files in Visual Studio.
3. To use the C++ code provided by the Online Services plugins, you must add the `OnlineServicesInterface` module as public dependencies to your project's **.Build.cs** file.
4. Open your project's **.Build.cs** file from the **Solution Explorer** by navigating to **Games > [YOUR_GAME] > Source > [YOUR_GAME] > [YOUR_GAME].Build.cs**.
5. Add `OnlineServicesInterface` and `CoreOnline` to your `.Build.cs` public dependencies. Your `.Build.cs` file should look like this:

```
// Copyright Epic Games, Inc. All Rights Reserved.

using UnrealBuildTool;

public class MyProject : ModuleRules
{
public MyProject(ReadOnlyTargetRules Target) : base(Target)
{
PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;
PublicDependencyModuleNames.AddRange(new string[] {
"Core",
"CoreUObject",
"Engine",
"InputCore",
"EnhancedInput",
"OnlineServicesInterface",
"CoreOnline"
});
}
}
```

Copy full snippet

6. Save your changes in Visual Studio.

## Generate Project Files

Since you made a change to your project's **.Build.cs** file, you need to refresh your Visual Studio project files. This ensures that the changes you made are reflected in Visual Studio Intellisense and lets you use the functionality in the plugins that you just added.

To generate project files, follow these steps:

1. Close Visual Studio.
2. Navigate back to your open project in the Unreal Editor.
3. Regenerate your Visual Studio project files by selecting **Tools > Refresh Visual Studio Project**.

A progress bar appears, displaying the status of the updates to your code project and disappears when the process is complete.

# Configure Default Platform Services

The last step is to specify your default platform services for the Online Services plugins. The default platform services specify which backend platform services are returned by a call to `UE::Online::GetServices`. A list of available platform identifiers is given on the Overview of Online Services documentation page.

To specify Online Services Null as the default platform services, follow these steps:

1. Open your project in Visual Studio. You can do this by navigating to **Tools > Open Visual Studio** from within the Unreal Editor.
2. Open your project's **DefaultEngine.ini** file in the Visual Studio Solution Explorer by navigating to **Games > [YOUR_GAME] > Config > DefaultEngine.ini**.
3. Add the following to your project's **DefaultEngine.ini** file:

```
[OnlineServices]
DefaultServices=Null
```

Copy full snippet

Online Services Null is an online services implementation that does not use any backend online services. This is used for testing and debugging your online services implementation without a backend service. If you want to use a different backend online service as your project's default online services, you can choose one from the list provided in the Configuration section of the Online Services Overview.

# Access Online Services in Your Project

The Online Services plugins are now enabled and configured for use in your project. To access the Online Services plugin and its various interfaces, follow these steps:

1. Add `#include "Online/OnlineServices.h"` to the file where you want to access the Online Services plugin.
2. Obtain a pointer to the default platform services with `IOnlineServicesPtr OnlineServicesPtr = UE::Online::GetServices();`.

Now you can access the different Online Services plugin interface functionalities. For example, to access the Auth Interface, follow these steps:

1. Make sure you have first obtained a pointer to the default platform services.
2. Add `#include "Online/Auth.h"` to the file where you want to access the Auth interface.
3. Obtain a pointer to the Auth interface with `IAuthPtr AuthPtr = OnlineServicesPtr->GetAuthInterface();`.

Now you can access the functionality of the Auth interface through the Auth interface pointer. The same logic works for all other Online Services interfaces.