

Epic Games Protocol Activation

AKA Deep-Linking

Overview	1
Technical Details	1
Usage	2
Examples	2
Launch Game PDP	2
Launch Add-On PDP	3
Launch Game	3
Making a protocol activation request	3
Checking to see if the EGS Launcher is installed	3
Best Practices	4
Testing protocol activation	4
See Also	5

Overview

The Epic Games Store (EGS) Launcher can be launched via protocol activation by game clients. Activation parameters can be used to launch directly to specific Product Details Pages (PDPs), a process also known as deep-linking. Game clients can request themselves or other game clients to be (re)launched directly.

All of these protocol activations can be accomplished using OS-provided functionality.

Protocol activation is particularly useful for titles that have an in-game store that do not want or are unable to integrate with the Purchase Overlay in the EOS SDK.

Protocol activation as a means to restart the game client is also a useful way to get a new `exchange_code` in cases where the game was not launched properly or the `exchange_code` has expired.

Technical Details

The EGS Launcher is registered via the OS to handle `com.epicgames.launcher` protocol activations. Any time a request is made to the OS to handle the `com.epicgames.launcher`

URI, the EGS Launcher will receive the request (assuming the EGS Launcher is installed). The EGS Launcher accepts additional parameters to further control how it is launched.

Usage

EGS Launcher protocol activation supports a number of launch options.

Parameters:

- `path`
 - Launch to a specific PDP or app on the store.
- `action`
 - `launch` - launches an app
 - `updatecheck` - get launcher to poll for available updates
 - `installer` - opens the installation options screen for the app if there are any
- `silent`
 - `true` - used with the `action=launch` parameter to launch the app in question without visibly popping up the EGS Launcher. The silent option is suggestive, if the launcher determines UI is required to complete the request, it will be shown.

Examples

Important Note: The specific strings to use for your app and offers are determined by back end configuration. Please work with the Service Delivery team to configure these appropriately.

The path portion of a store URI will match the product slug that is available on the web. For example:

[https://www.epicgames.com/store/product/\[product slug\]](https://www.epicgames.com/store/product/[product slug]) →

```
com.epicgames.launcher://store/product/[product slug]
```

The path portion of an app's URI is a combination `Sandbox ID` (also called `Namespace ID`), `Catalog ID` (also called `Item ID`), and `Artifact ID`, Separated by URL-encoded colons.

For example:

```
com.epicgames.launcher://apps/[SandboxID]%3A[CatalogID]%3A[ArtifactID]?action=launch&silent=true
```

You might encounter a deprecated format that uses only `Artifact ID` and is still supported for backwards compatibility.

Alternatively, instead of the `SandboxID:CatalogID:ArtifactID` for the app's URI, a url encoded installation path to the app's install directory on the machine (or any location inside of

it) may be used. This may be easier for game client code to construct, particularly when it is used across multiple different ArtifactIds. For example:

```
com.epicgames.launcher://apps/D%3A%2FGames%2FMyGame?action=launch&silent=true
```

Launch Game PDP

To launch to the Product Details Page in the store for a specific game:

```
com.epicgames.launcher://store/product/fortnite
```

Launch Add-On PDP

To launch to the Product Details Page in the store for an add-on beneath a game:

```
com.epicgames.launcher://store/product/fortnite/vox-hunters-quest-pack
```

Launch Game

To launch the game itself while moving the EGS Launcher to the background:

```
com.epicgames.launcher://apps/fn%3A4fe75bbc5a674f4f9b356b5c90567da5%3AFortnite?action=launch&silent=true
```

Making a protocol activation request

Windows supports a number of ways to initiate a protocol activation request.

- From a command line. Use the start command with the appropriate URI. For example:
 - `start com.epicgames.launcher://store/product/fortnite`
- From the shell. Windows+R, place the URI in the “Open:” text field, and press Enter. For example:
 - `com.epicgames.launcher://store/product/fortnite`
- Within an application, via OS provided APIs. C++ example:

```
const char* uriA =  
"com.epicgames.launcher://store/product/fortnite";  
wchar_t uriW[ MAX_PATH ];  
std::copy( uriA, uriA + lstrlenA( uriA ) + 1, uriW );  
::ShellExecuteW( NULL, L"open", uriW, NULL, NULL, SW_SHOWNORMAL  
);
```

On Mac, you can run this from a terminal window or using Spotlight Search

- From a terminal window, use the open command with the appropriate URI. For example:
 - `open com.epicgames.launcher://store/product/fortnite`
- For Spotlight Search, paste the URI in the search text field, and press Enter. For example:
 - `com.epicgames.launcher://store/product/fortnite`

Checking to see if the EGS Launcher is installed

Before making an API call to protocol activate the EGS Launcher, it is advisable to validate that the EGS Launcher is installed.

On Windows, this can be done by checking the registry. C++ example:

```
bool bRes = false;

HKEY hKey;
if (RegOpenKeyEx(HKEY_CLASSES_ROOT,
TEXT("com.epicgames.launcher"), 0, KEY_READ, &hKey) ==
ERROR_SUCCESS)
{
    RegCloseKey(hKey);
    bRes = true;
}

return bRes;
```

On Mac, this can be done by checking for a registered URL Scheme. Objective-C example:

```
BOOL Result = [[UIApplication sharedApplication]
canOpenURL:[NSURL URLWithString:@"com.epicgames.launcher:"]];
```

Best Practices

Testing protocol activation

Testing protocol activation end-to-end requires a number of steps:

1. Get a store-capable Product ID and associated upload credentials
2. Write the game client code
3. Upload the game client using BPT or Dev Portal
4. Configure product slugs and other associated strings
5. Create a PDP
6. Install the EGS Launcher
7. Acquire the audience for the game (usually via key)
8. Install the game
9. Make a protocol activation request

Instead, it's possible to unit test certain aspects of protocol activation with fewer steps:

- To test the URI itself, simply use the Windows+R run command with the desired protocol activation and parameters.
- To test game client code, consider pointing to an existing offer or game instead. It can be any game in the EGS Launcher.

See Also

Check out the latest documentation for:

- Content ingestion
- Client authentication
- Purchase Overlay