

HttpLibrary Plugin

The HttpLibrary plugin supports Windows, Mac, Linux, Android, and iOS. Also starting with version **4.19** you must **download and install the JsonLibrary plugin** since it is required.



DOWNLOAD

This plugin can be downloaded from GitHub at the following address:

<https://github.com/tracerinteractive/UnrealEngine/releases>

A screenshot of a GitHub release page for version 4.19.0. The page shows the version number '4.19.0' in large blue text, with a 'Latest release' badge to its right. Below the version number, it shows the commit hash '8e4560b'. The release was made by 'tracerinteractive' 9 hours ago. Under the 'Assets' section, there are three zip files listed: 'HttpLibrary-4.19.zip' (20.4 MB), 'JsonLibrary-4.19.zip' (19.3 MB), and 'WebUI-4.19.zip' (17.9 MB).

Asset Name	Size
HttpLibrary-4.19.zip	20.4 MB
JsonLibrary-4.19.zip	19.3 MB
WebUI-4.19.zip	17.9 MB

Each version is also compatible with all minor engine updates which means the 4.19.0 version of the plugin will work with any corresponding hotfix such as 4.19.1 or 4.19.2 as well.

You must have a GitHub account linked to your Epic Games account!

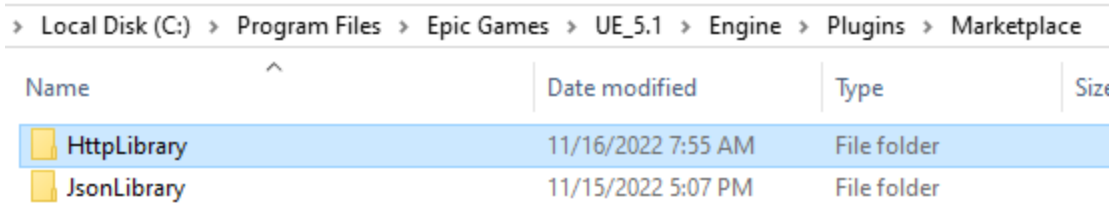
Setup Instructions: <https://www.unrealengine.com/ue4-on-github>



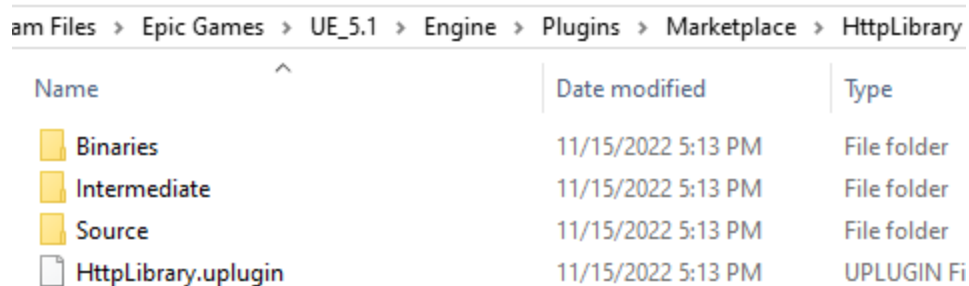
Otherwise you will receive the previous 404 error if you are not signed in with a linked account.

INSTALL

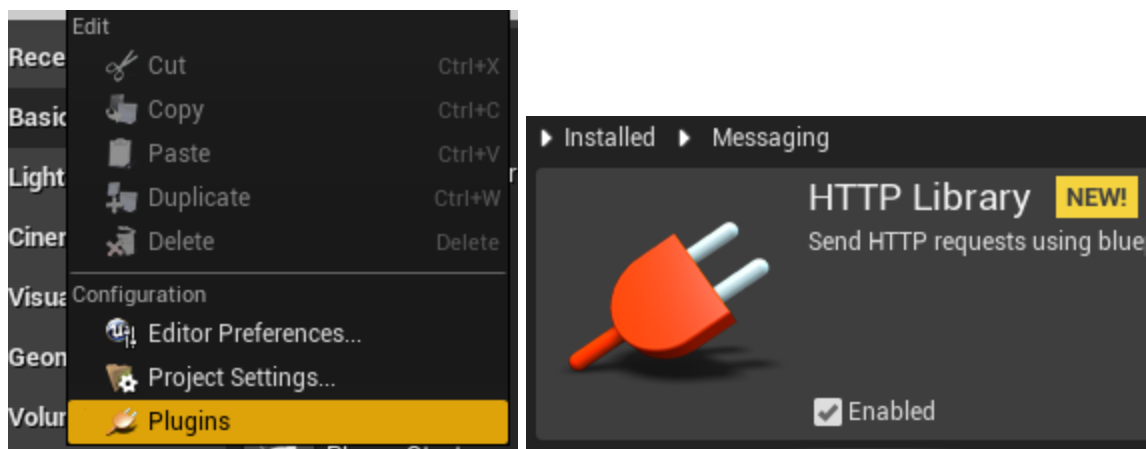
To install the HttpLibrary plugin extract the downloaded files to the following engine folder:



Also take note of the **UE_5.1** directory in the screenshots. You need to change this folder to the version that corresponds to the plugin version that was downloaded. *If you did not install your engine to the default directory then navigate to your custom installation folder instead.*



Then open your project and go to the “Plugins” option in the edit drop-down. Click on the “Messaging” category and enable the HttpLibrary plugin if it is not already enabled.



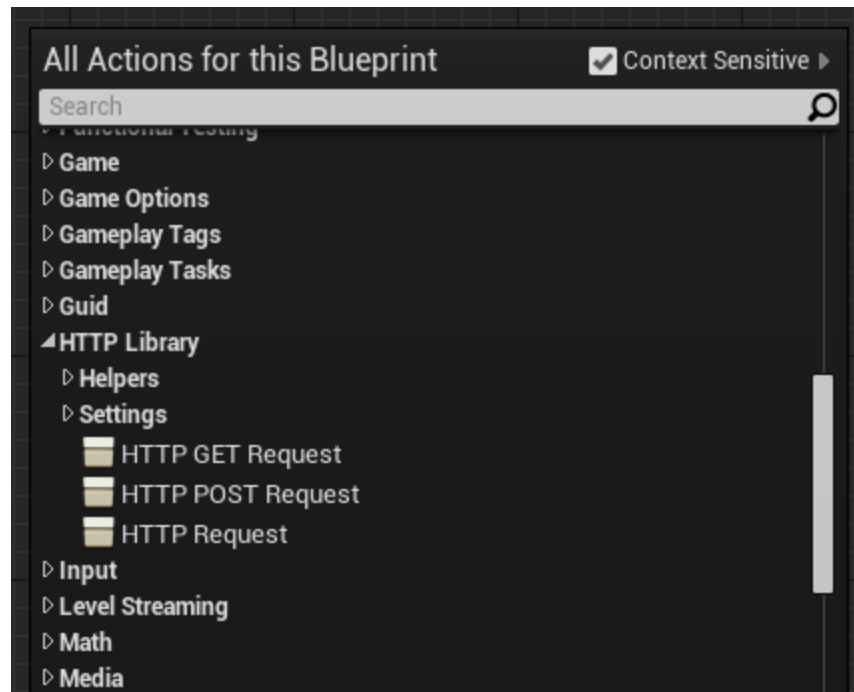
You have now successfully installed the HttpLibrary plugin. Restart the editor to continue.

TABLE OF CONTENTS

MENU	4
GET	5
POST	6
QUERY STRING	7
REQUEST	8
SETTINGS	10
COMPILE	11

MENU

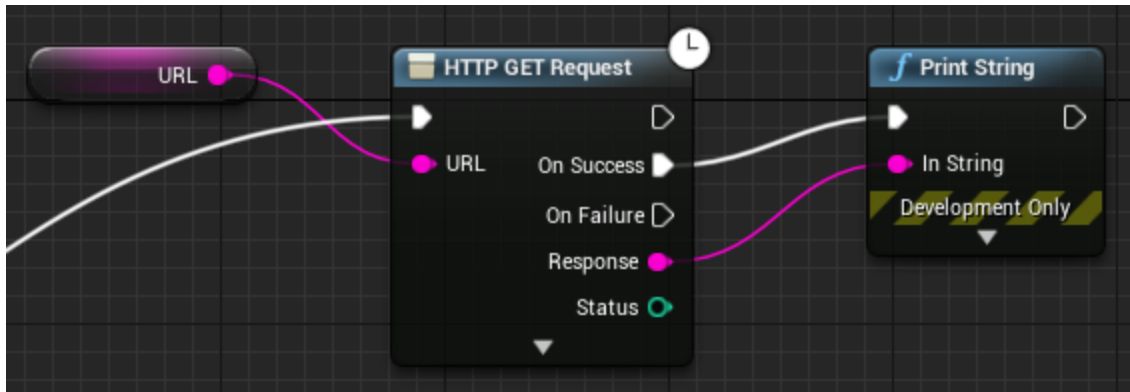
Once the HttpLibrary plugin is installed and enabled start by right clicking in the event graph of any blueprint and navigating to the *HTTP Library* section.



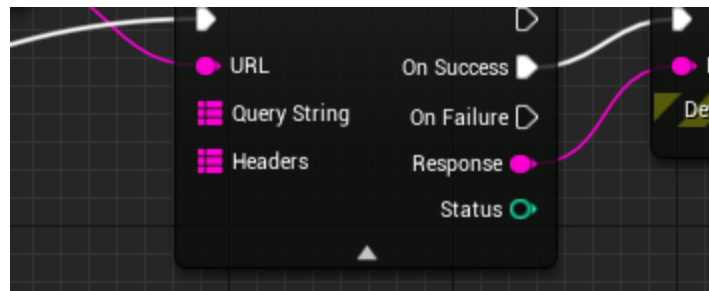
Note that this plugin uses **latent nodes** (which is signified by the **clock symbol**) and therefore cannot be added to blueprint functions. These nodes must exist on the event graph like any other custom events due to their asynchronous functionality.

GET

The “HTTP GET Request” node is a simple way to request data. Just provide a URL and receive either the response text or the HTTP status code if the request failed.

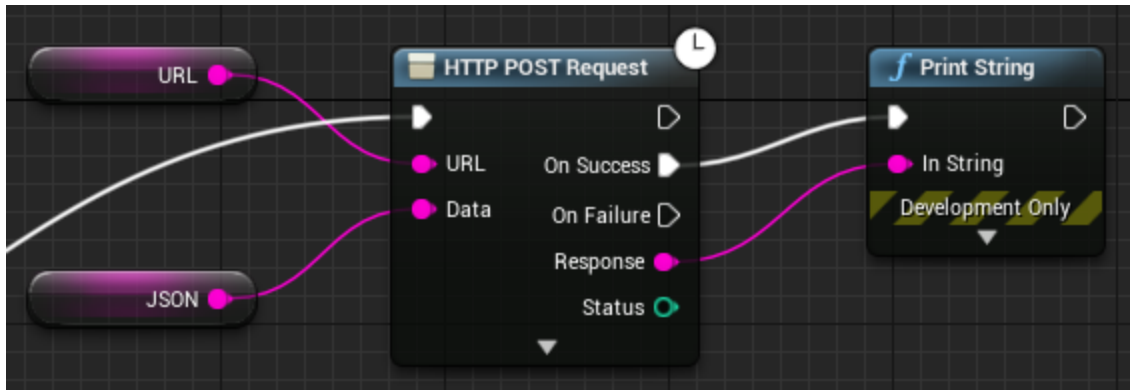


Click the **advanced display** (which is the arrow on the bottom) to view more options. This provides the ability to specify custom headers in the request along with appending any custom query string parameters to the end of the URL.

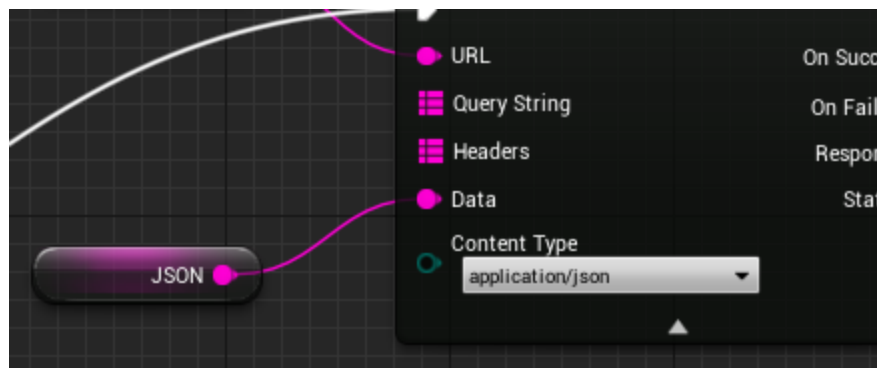


POST

The “HTTP POST Request” node is a simple way to send data. Just provide a URL along with any data being sent. Then receive either the response text or the HTTP status code if the request failed.



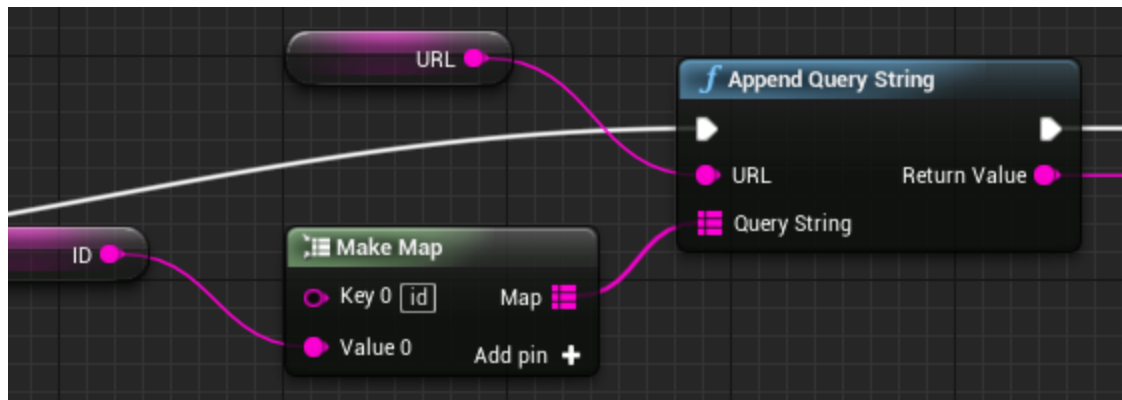
Click the **advanced display** to view more options. This provides the ability to specify the content type of the data being sent along with any custom headers in the request.



Note that the default content type is text/plain but may be changed to application/json, application/xml, or many other common MIME types.

QUERY STRING

The “Append Query String” node provides explicit access to the same functionality as the *query string* pin shown previously in the advanced display. Just provide the URL along with a string map of the parameters.

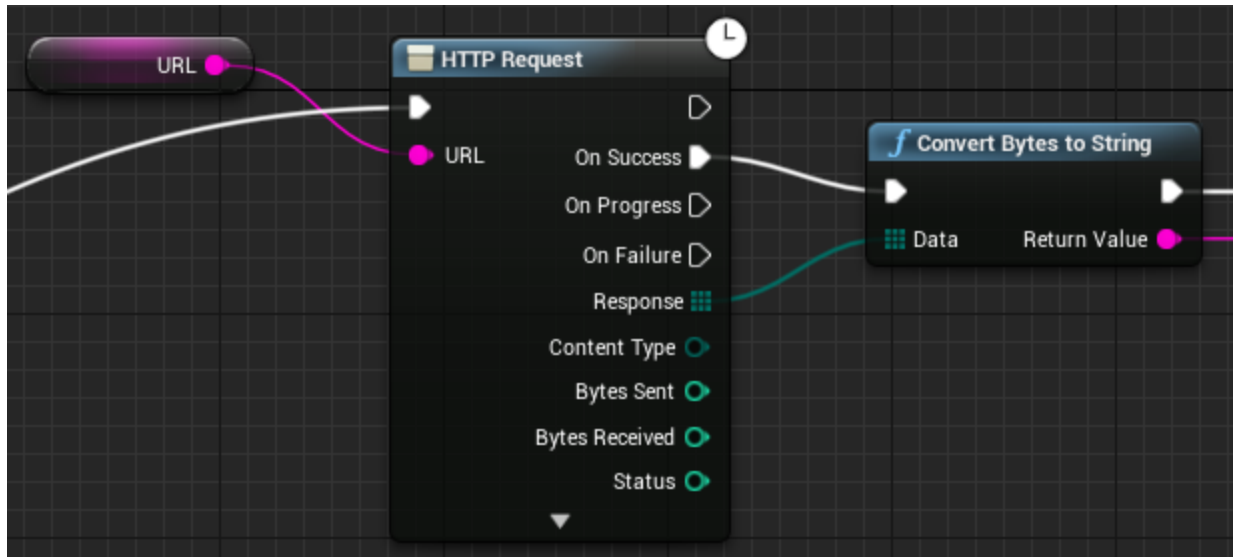


This function will **automatically detect if a query string exists** on the provided URL. If no query string was found a new one will be appended starting with `?key0=value0&key1=value1`.

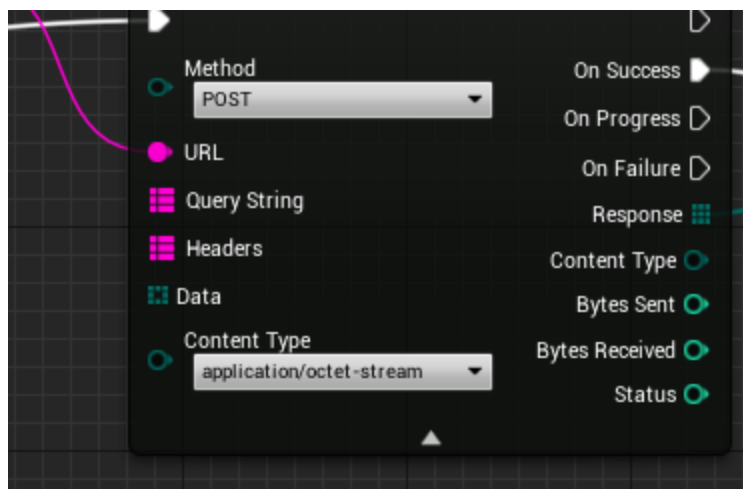
However if a query string was already found on the URL then an existing one will be appended starting with `&key0=value0&key1=value1` instead. This provides the ability to append custom parameters without having to worry about the context of the input URL.

REQUEST

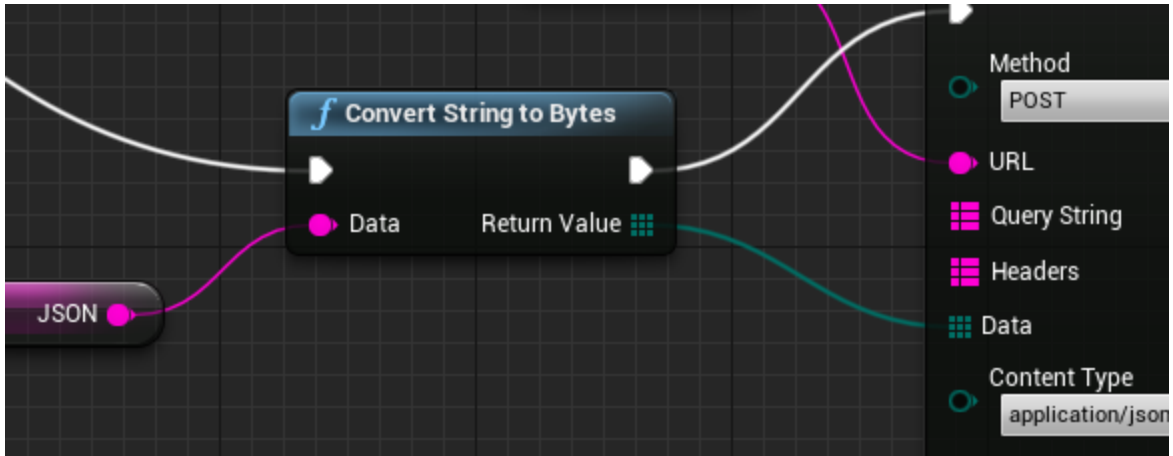
The “HTTP Request” node is an advanced way to send and receive data. Start by providing a URL and receive raw binary data along with the content type of the response. The “Convert Bytes to String” helper function is also available to convert this raw binary data into a string.



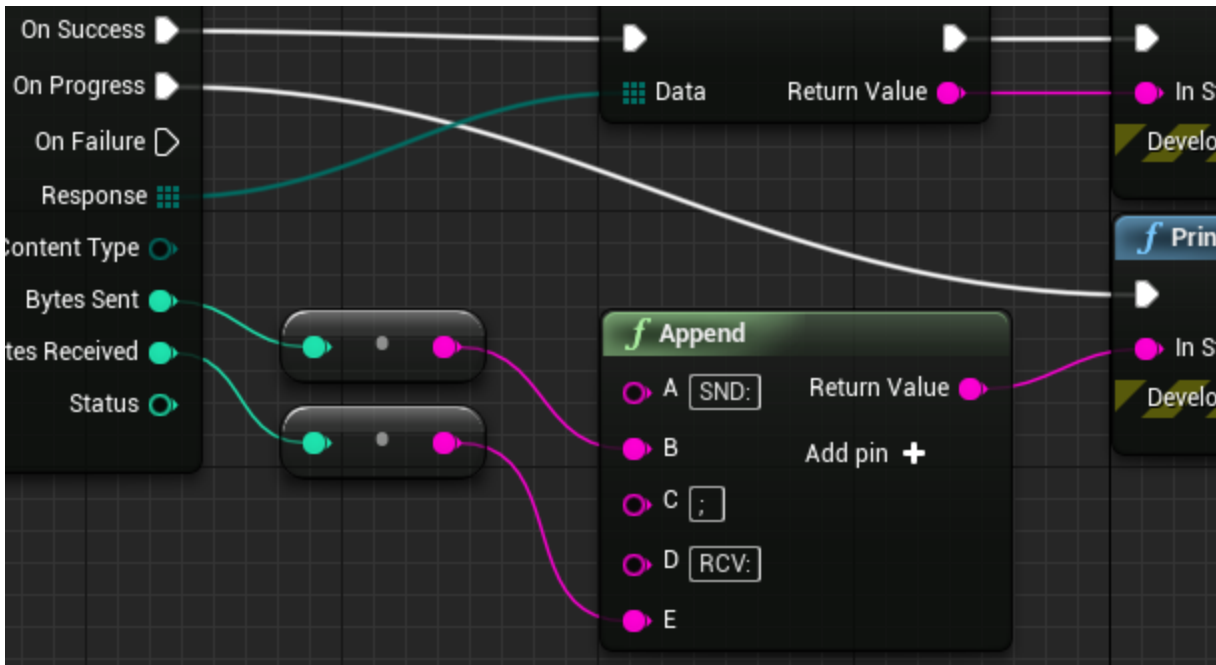
Click the **advanced display** to view more options. This allows you to specify the HTTP request method along with the content type of the raw binary data being sent.



The “Convert String to Bytes” helper function is also available to convert a string into raw binary data before sending.



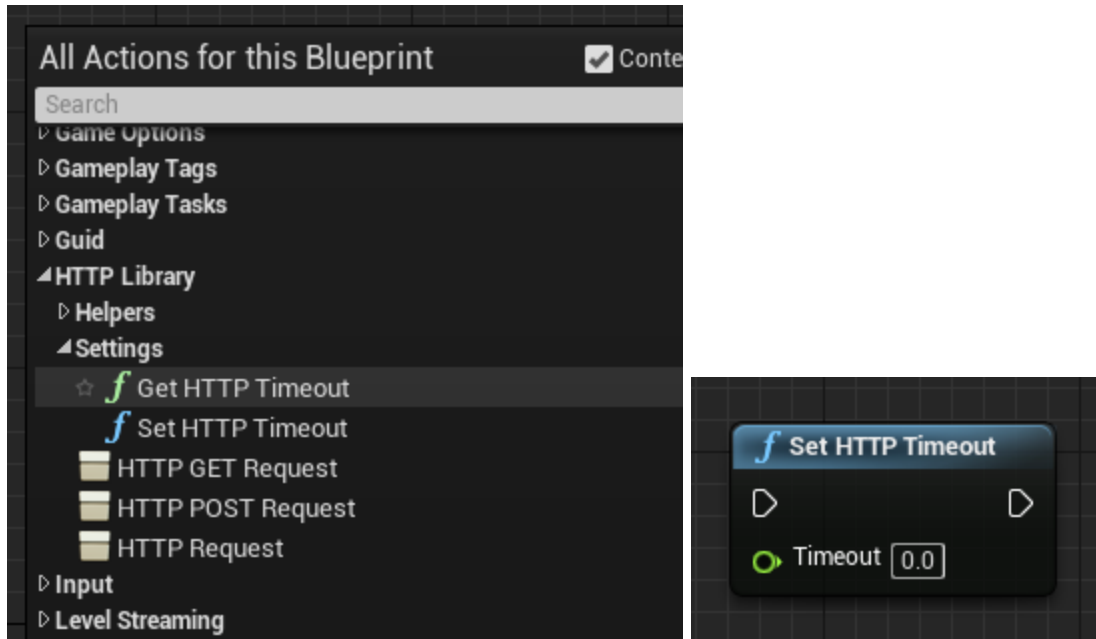
The "HTTP Request" node also has another execution pin for progress updates. This provides the ability to monitor the number of bytes that are sent and received during the request.



Note that only the *Bytes Sent* and *Bytes Received* pins are accessible during these progress updates. The *Response*, *Content Type*, and *Status* pins are only available on execution of the success or failure callbacks, respectively.

SETTINGS

The “Get HTTP Timeout” and “Set HTTP Timeout” nodes are available in the *Settings* section to customize the timeout settings of the engine.



There are also helper functions that allow you to convert the content type enum into a header string and vice versa. These are located in the *Helpers* section of the *HTTP Library* menu.

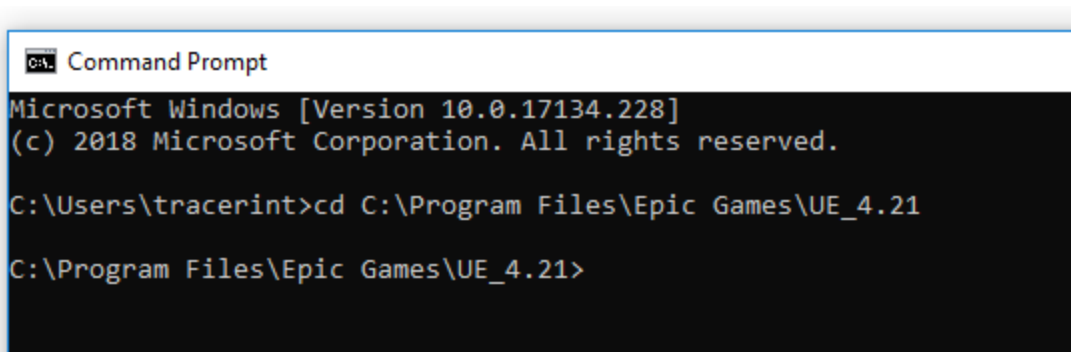
COMPILE

This plugin can be manually compiled for other platforms or engine versions. First open the command prompt (by searching for “cmd” in the start menu) and type the following command:

```
cd "C:\Program Files\Epic Games\UE_4.21"
```

You can copy this command and paste it by right-clicking on the command prompt. Also take note of the **UE_4.21** directory. You need to change this folder to the version that corresponds to the engine version you are using. *If you did not install your engine to the default directory then type the path to your custom installation folder instead.*

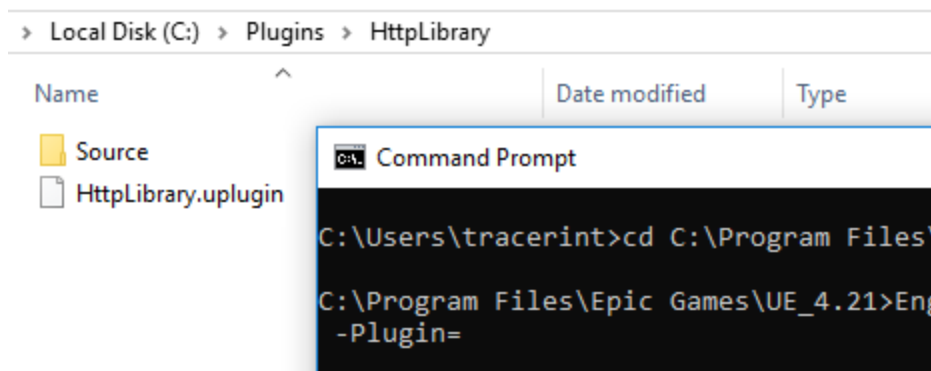
Press ENTER to run the command. You should now see output similar to the following:



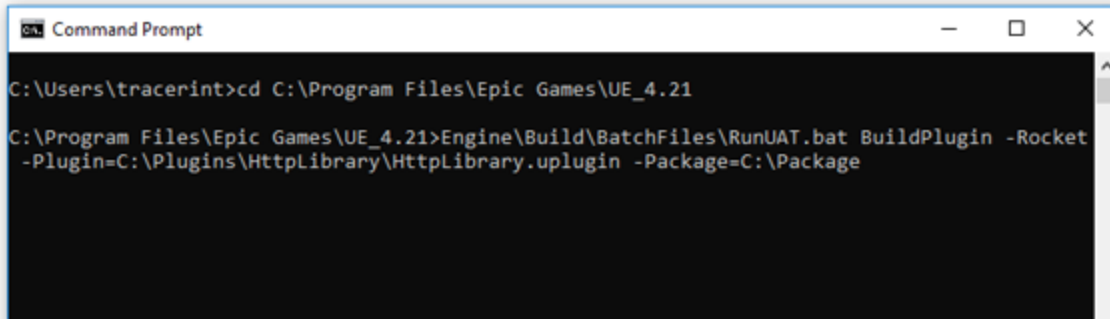
Then type the following command to build the plugin:

```
Engine\Build\BatchFiles\RunUAT.bat BuildPlugin -Rocket -Plugin="..." -Package="..."
```

Replace the first "..." with the path to HttpLibrary.uplugin and the last "..." with the path to a temporary “package” folder. You can also drag and drop the .uplugin file or your temporary folder directly onto the command prompt and it will automatically type the path:

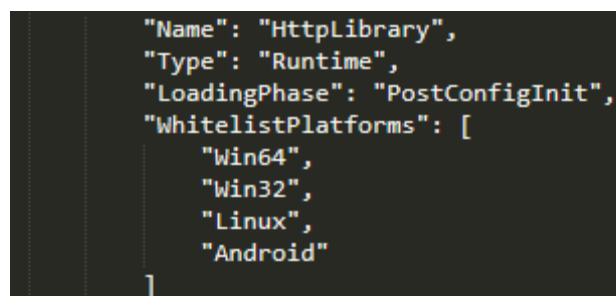


Make sure these paths are **not inside the engine directory** or the build will fail. Once you have the full command typed out it should look similar to the following:



```
Command Prompt
C:\Users\tracerrint>cd C:\Program Files\Epic Games\UE_4.21
C:\Program Files\Epic Games\UE_4.21>Engine\Build\BatchFiles\RunUAT.bat BuildPlugin -Rocket
-Plugin=C:\Plugins\HttpLibrary\HttpLibrary.uplugin -Package=C:\Package
```

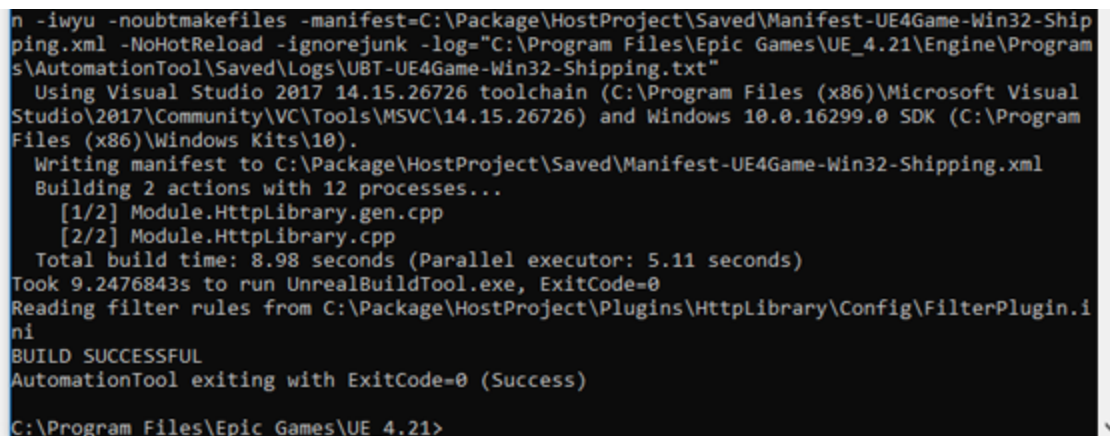
If your machine does not support Mac or Linux builds then you will most likely have to remove the “Mac” and “IOS” or “Linux” platforms from HttpLibrary.uplugin before compiling:



```
{
  "Name": "HttpLibrary",
  "Type": "Runtime",
  "LoadingPhase": "PostConfigInit",
  "WhitelistPlatforms": [
    "Win64",
    "Win32",
    "Linux",
    "Android"
  ]
}
```

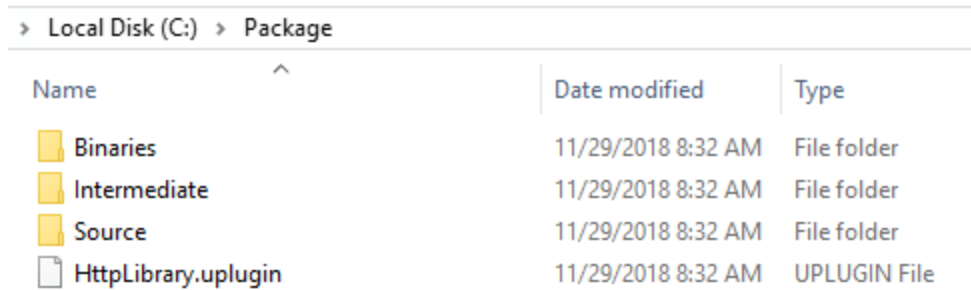
Now hit the ENTER key to run the command. If everything was setup correctly you’ll see many different versions of the plugin being compiled for various platforms.

Once the build is complete you should see the “BUILD SUCCESSFUL” message:



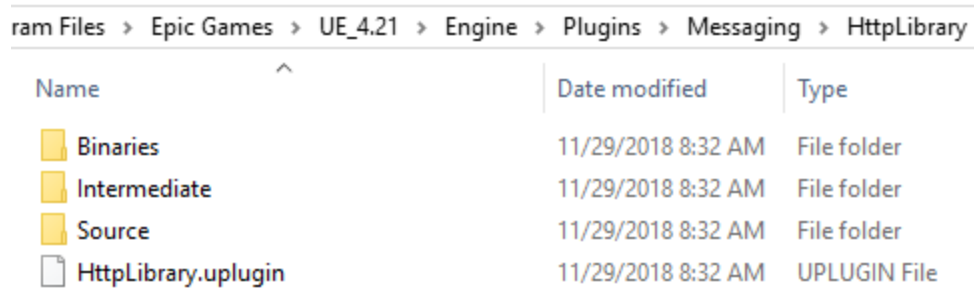
```
n -iwyu -noubtmakefiles -manifest=C:\Package\HostProject\Saved\Manifest-UE4Game-Win32-Shipping.xml -NoHotReload -ignorejunk -log="C:\Program Files\Epic Games\UE_4.21\Engine\Programs\AutomationTool\Saved\Logs\UBT-UE4Game-Win32-Shipping.txt"
Using Visual Studio 2017 14.15.26726 toolchain (C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\VC\Tools\MSVC\14.15.26726) and Windows 10.0.16299.0 SDK (C:\Program Files (x86)\Windows Kits\10).
Writing manifest to C:\Package\HostProject\Saved\Manifest-UE4Game-Win32-Shipping.xml
Building 2 actions with 12 processes...
[1/2] Module.HttpLibrary.gen.cpp
[2/2] Module.HttpLibrary.cpp
Total build time: 8.98 seconds (Parallel executor: 5.11 seconds)
Took 9.2476843s to run UnrealBuildTool.exe, ExitCode=0
Reading filter rules from C:\Package\HostProject\Plugins\HttpLibrary\Config\FiltersPlugin.ini
BUILD SUCCESSFUL
AutomationTool exiting with ExitCode=0 (Success)
C:\Program Files\Epic Games\UE_4.21>
```

Check your temporary “package” folder to ensure it looks similar to the following:



Name	Date modified	Type
Binaries	11/29/2018 8:32 AM	File folder
Intermediate	11/29/2018 8:32 AM	File folder
Source	11/29/2018 8:32 AM	File folder
HttpLibrary.uplugin	11/29/2018 8:32 AM	UPLUGIN File

Then copy the files in this temporary folder into your engine installation directory. You must do this beforehand if you are compiling any other plugins that require the HttpLibrary plugin.



Name	Date modified	Type
Binaries	11/29/2018 8:32 AM	File folder
Intermediate	11/29/2018 8:32 AM	File folder
Source	11/29/2018 8:32 AM	File folder
HttpLibrary.uplugin	11/29/2018 8:32 AM	UPLUGIN File

You have now successfully compiled the HttpLibrary plugin for your engine version.