# Sending HTTP requests

*This article requires basic knowledge of Lua, RoCRAFTERO Studio, and how to use the Internet.*

[HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService) allows you to send and receive information from websites. The most common use for this is *external tools*, which are basically programs that are not part of RoCRAFTERO Studio. However, they can be used for more than just that.

HttpService starts off as disabled by default. HttpService can be enabled through the the Game Settings button in Studio.

GameSettings.PNG AllowHTTPRequests.PNG

**Contents**

* [1](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#How_it_works)
* [How it works](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#How_it_works)
  + [1.1](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#GetAsync)
  + [GetAsync](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#GetAsync)
  + [1.2](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#PostAsync)
  + [PostAsync](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#PostAsync)
* [2](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#Usage)
* [Usage](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#Usage)
* [3](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#Restrictions)
* [Restrictions](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#Restrictions)
* [4](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#Common_errors)
* [Common errors](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#Common_errors)
  + [4.1](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#HTTP_requests_are_not_enabled)
  + [HTTP requests are not enabled](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#HTTP_requests_are_not_enabled)
  + [4.2](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#Http_requests_can_only_be_executed_by_game_server)
  + [Http requests can only be executed by game server](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#Http_requests_can_only_be_executed_by_game_server)
  + [4.3](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#trust_check_failed)
  + [trust check failed](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#trust_check_failed)
* [5](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#See_also)
* [See also](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Sending_HTTP_requests#See_also)

## How it works

When you visit a website on your web browser, you are making something called an HTTP Request. Your computer says to the website, *"Hey, could I have this page?"* The website then looks for (or creates, in some cases) the correct information, and sends it to your computer. Your computer then sorts through that data, and you get a nice little graphic on your screen.

Now, your typical run-of-the-mill webpage will give you information in something called *HTML*. HTML (short for HyperText Markup Language) looks a little something like this:

'"`UNIQ--nowiki-00000007-QINU`"'

Looks confusing, right? Wrong. HTML is pretty much basic English. With minimal knowledge and some common sense, you can read it. With Google, you can write it. To see some basic HTML, press F12 on your keyboard and go to the "Elements" tab. However, this isn't a tutorial about HTML formatting.

Back on topic, webpages can also give you other information. For example, the webpage [http://www.roblox.com/game/GetCurrentUser.ashx](https://web.archive.org/web/20180324102307/http://www.roblox.com/game/GetCurrentUser.ashx) gives you the user ID of the person who is logged in, but in plain text. Now let's look at some more specific things.

### GetAsync

[GetAsync](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/HttpService/GetAsync) fires your typical web request, which is a *get* request. Every time you visit a website, you are sending a request with the command *get*[unconfirmed]. The web server sees this, and takes it as "send me information!" This is what [GetAsync](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/HttpService/GetAsync) does. When you call [GetAsync](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/HttpService/GetAsync), RoCRAFTERO sends a request. Your script will yield until either a) the request is completed or b) the request times out. A timeout happens when a server goes a certain amount of time without a response.

*--Simple Example of a GET request sent to example.com*

hs = game:GetService("HttpService")

test = hs:GetAsync("http://example.com", true)

*-- test now contains the html text representation of example.com.*

If you are trying to access the same information from a studio instance, you must be in Server mode (Tools -> Start Server). After the above example you can tell Lua to print(test) and you will be able to see the HTML contents of *example.com*.

### PostAsync

When you update your status on Roblox, you're sending something called a *post* request. A post request holds information, along with the tag *post*[unconfirmed]. This is what [PostAsync](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/HttpService/PostAsync) does. When you call [PostAsync](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/HttpService/PostAsync), your script will yield until either a) the request is completed or b) the request times out.

*-- THIS EXAMPLE WILL FAIL, but this is technically how you make a post request to example.com if you had the correct permissions.*

hs = game:GetService("HttpService")

myContent = "This\_is\_a\_pointless string that means nothing, but could have useful content."

hs:PostAsync("http://example.com", myContent)

## Usage

Turning on [HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService) is quite easy. In RoCRAFTERO Studio, look at your top menu bar. Press Insert -> Service -> [HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService). Once you locate [HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService) under the Explorer window, find [HttpEnabled](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/HttpService/HttpEnabled) under Properties. Set this to true (check the box), and you're ready to roll!

It is also important to know how to use [JSONEncode](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/HttpService/JSONEncode) and [JSONDecode](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/HttpService/JSONDecode) methods, as content being posted to HTTP is usually in the format of a string.

[HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService) has millions of usages. You can use it to do things like create a web browser, or make a global leaderboard. (But maybe you should use [OrderedDataStores](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/OrderedDataStore) for that, since that's easier and doesn't need HttpService) Once you have a webserver set up, you can use [HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService) to connect to webpages on it. An example of this would be RoCRAFTERO [web API](https://web.archive.org/web/20180324102307/http://api.roblox.com/). You can connect to those webpages using [HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService), and find lots of cool stuff!

The following is a made-up example of using HTTP Service to post a Lua Table to [http://example.org/json-table](https://web.archive.org/web/20180324102307/http://example.org/json-table)

myTable = {}

myTable["exp"] = 1100

myTable["gold"] = 999

myTable["swordID"] = 1000398

*-- Initialize an HTTP Service pointer*

hs = game:GetService("HttpService")

*-- Encodes the Table into a JSON string.*

myTableJSON = hs:JSONEncode(myTable)

*-- Posts the JSON string to http://example.org/json-table*

hs:PostAsync("http://example.org/json-table", myTableJSON)

Retrieving the same table after it has been posted, and assuming that the previously mentioned URL Contained NOTHING ELSE BUT the JSON table.

*-- Initialize an HTTP Service pointer*

hs = game:GetService("HttpService")

*-- Uses a GET Request to pull down whatever is on http://example.org/json-table, which in this example is simply the JSON table posted previously.*

myTableJSON = hs:GetAsync("http://example.org/json-table", true)

*-- Decodes the JSON String we just received into a Lua Table.*

myTable = hs:JSONDecode(myTableJSON)

*-- Returns myTable, which looks like {"exp"=1100, "gold"=999, "swordID" = 1000398}*

## Restrictions

* [HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService) cannot access Roblox domains.
* You can only make 500 requests per minute. This is to stop players from clogging up the RoCRAFTERO servers.
* You must have "http://" or "https://" at the beginning of your URL.

## Common errors

### HTTP requests are not enabled

Locate [HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService) under the Explorer window, and find [HttpEnabled](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=API:Class/HttpService/HttpEnabled) under Properties. This has to be set to true (the box has to be checked) for [HttpService](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=HttpService) to work.

### Http requests can only be executed by game server

Http Requests can only be sent from a game server. These include:

* Your game in play mode
* Tools -> Test -> Start Server

Run your [Script](https://web.archive.org/web/20180324102307/http://wiki.roblox.com/index.php?title=Script) from there instead.

**PLEASE NOTE:** USE THIS URL FOR SAVING TABLES FOR TESTING DUE TO A JSON BUG:

<https://sitetest3.rocrafteroprod1.x10host.com/JSON%20STUDIO%20TESTING/json-table.json>