

The Wayback Machine - https://web.archive.org/web/20120911104147/http://www.wiremod.com/forum/expression-2-discussion-help/22376-e2-http-functions-usage-good-practices.html

WIREMOD

Sign in through STEAM

User Name

Password

Log in

Help

Register

☐ Remember Me?

What's New?

Articles

Forum

Blogs

Forum Home

New Posts

FAQ

Calendar

Community

Forum Actions

Quick Links

Advanced Search

Forum

Tool Help & Discussion

Expression 2 Discussion & Help

E2 HTTP functions - Usage and good practices.

If this is your first visit, be sure to check out the **FAQ** by clicking the link above. You may have to **register** before you can post: click the register link above to proceed. To start viewing messages, select the forum that you want to visit from the selection below.

If you'd like to receive bonuses in the Official Wiremod Gmod Server, Link your Steam account to you forum account [here!](#)

Results 1 to 7 of 7

Thread: E2 HTTP functions - Usage and good practices.

LinkBack

Thread Tools

Display

09-03-2010

#1

Filipe

Mario, hard at work

MEMBER

FILL PIPE

Join Date: Jun 2007

Location: Portugal - Lisbon

Posts: 401

Blog Entries: 1

E2 HTTP functions - Usage and good practices.

E2 HTTP functions allow you to retrieve online content with the HTTP protocol - a website (markup), .txt file, rss feed, etc.

This opens new doors for all sorts of inventive, fun and dynamic expressions, but it's also your task as the coder to use them responsibly.

Usage:

Doing an HTTP request is (roughly) a 6-step process:

1. You set **runOnHTTP(1)** on your **first()** block statement.

Example:

Code:

```
if(first()) {
    runOnHTTP(1)
}
```

2. You check if you can make a request with **httpCanRequest()**

httpCanRequest() returns a **number** that tells you if you're allowed to make a new request at this moment, there's a **default 3 second delay** between HTTP requests to prevent abuse.

Example:

Code:

```
if(httpCanRequest()) {
    httpRequest("www.wiremod.com")
} else {
    print("Cannot request www.wiremod.com !")
}
```

```
}
```

3. You make the request with `HttpRequest(string URL)`

HttpRequest(string URL) takes a **string** of the **URL (address)** to request, the 'http://' prefix is optional. This function does not return anything.

Example:

Code:

```
HttpRequest("http://www.wiremod.com")
```

4. You check that your request is ready with `HttpClk()`

HttpClk() returns a **number** (1) when the current execution was triggered by a request being completed.

Example:

Code:

```
if(HttpClk()) {  
    print("Request complete!")  
}
```

5. (OPTIONAL) You check what URL request was completed with `HttpRequestUrl()`

HttpRequestUrl() returns a **string** with the URL of the request that was just completed.

Example:

Code:

```
if(HttpClk()) {  
    RequestURL = HttpRequestUrl()  
    print("Request complete: "+RequestURL)  
}
```

6. The most important part, you retrieve the request's data, that is, a string with the contents of the page/file you requested, with `HttpData()`

HttpData() returns a **string** with the content you fetched.

Example:

Code:

```
if(HttpClk()) {  
    RequestURL = HttpRequestUrl()  
    print("Request complete: "+RequestURL)  
  
    Data = HttpData()  
    fileWrite("myfile.txt", Data)  
    print("Data saved!")  
}
```

Notes and good practices:

1. HTTP requests are *costly*, and can take a *considerable time to complete* - you should avoid making too many consecutive requests, and above all, avoid redundant ones. The best way to prevent the latter is *caching*.

Note, this is an non-functional E2, and servers only as a proof of concept.

Example:

Code:

```
@name CacheTest
@persist CACHE:table

# Note - this is an example, and not fully functional.
if(first()) {
    runOnHTTP(1)
}

URL = "www.wiremod.com"
if(CACHE[URL, string]) {
    # We have this page stored already.
    Page = CACHE[URL, string]
} else {
    if(httpCanRequest()) {
        httpRequest(URL)
    } else {
        print("Cannot request at this time.")
        # Consider using a buffer to account for
        # limits, without losing requests.
    }
}

if(httpClk()) {
    GotURL = httpRequestUrl()
    Data = httpData()
    CACHE[GotURL, string] = Data
    print("Saved "+GotURL+" to the cache.")
}
```

2. Strings retrieved from *httpData()* can be too huge to print.

Attempting to print a full string retrieved with *httpData()* may cause you to reach the *umsg* length limit. You can preview these strings by clamping their size:

Example:

Code:

```
if(httpClk()) {
    GotURL = httpRequestUrl()
    Data = httpData()
    print(Data:left(255))
}
```

3. HTTP requests can *timeout*, if the target address cannot be reached. If *httpData()* returns an empty string, an error may have occurred.

Note, this is an non-functional E2, and serves only as a proof of concept.

Example:

Code:

```
if(httpClk()) {
    GotURL = httpRequestUrl()
    Data = httpData()
    if(Data != "") {
```

Code:

```
print("Request complete!")
} else {
    print("An error occurred!")
}
}
```

4.If you're sending arbitrary strings as part of the URL, use the **httpUrlEncode(string)** function to properly encode it. The opposite can be achieved with **httpUrlDecode()**

Example:

Code:

```
MyString = "I do rather enjoy that smell, old chap!"
EncodedString = httpUrlEncode(MyString)
print(EncodedString)
# Output: I+do+rather+enjoy+that+smell%2C+old+chap%21
```

Snippets/Examples:

Building a GET key/value pair string dynamically:

Code:

```
# Our base target url:
URL = "http://www.somesite.com/killanator.php"

# Our table with arguments and values:
GET = table()
GET["name", string] = "Filipe"
GET["location", string] = "Portugal"
GET["isalien", string] = "false"

# 'Buffer' will hold our final key/value pairs string.
Buffer = ""
foreach(K, V:string = GET) {
    Buffer += format("%s=%s&", K, V)
}

# Some concatenation and...
RequestUrl = URL + "?" + Buffer
# Ta-ta!
print(RequestUrl)
# Output:
# http://www.somesite.com/killanator.php?name=Filipe&location=Portugal&isalien=false&
```

Buffering and dynamically storing requests:

Code:

```
@persist STORAGE:table

if(first()) {
    # For demonstration purposes only, an array
    # with various URLs that we'll load in sequence.
    Requests = array(
        "www.wiremod.com",
        "www.nasa.gov",
    )
}
```

```
        "www.mysite.com"
    )

    runOnHTTP(1)
}

# If we call the retryRequest timer, or if this is
# the first execution...
if(c1k("retryRequest") | first() ) {

    # If httpCanRequest() returns 1 we remove the first(bottom) url
    # from the queue, and call an httpRequest() with it.
    # Otherwise, the URL is still left in it's place, and the
    # buffer will attempt the request again the next time
    # it's executed.

    if(httpCanRequest()) {
        RequestUrl = Requests:removeString(1)
        httpRequest(RequestUrl)
    } else {
        timer("retryRequest", 2000)
    }
}
```

Last edited by Filipe; 09-03-2010 at 07:43 PM.

Share

beep boop

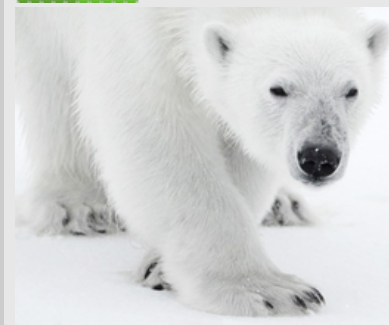
Reply With Quote

09-03-2010

#2

Drunkie

Ursus maritimus



Join Date: Feb 2009
Location: Canada
Posts: 6,255
Blog Entries: 1

Re: E2 HTTP functions - Usage and good practices.

Good tutorial. Explains everything well and is kept simple.

Share

Reply With Quote

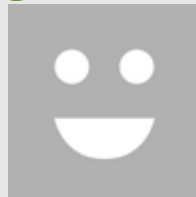
09-03-2010

#3

VVoMan_PL

Re: E2 HTTP functions - Usage and good practices.

Wire Noob



Join Date: Oct 2008

Location: Toruń, Poland

Posts: 15

Great tutorial! Finally got those functions in official SVN 🙏

Share

Reply With Quote

09-03-2010

#4

Filipe

Mario, hard at work

WIREMOD MEMBER



Join Date: Jun 2007

Location: Portugal - Lisbon

Posts: 401

Blog Entries: 1

Re: E2 HTTP functions - Usage and good practices.

Added two examples/snippets:

- Building a key/value string for GET requests from an arbitrary table.
- Dynamic request buffering and storage.

Share

Reply With Quote

09-03-2010

#5

Azrael

Wire Sofaking

WIREMOD MEMBER



Join Date: Aug 2007

Posts: 1,946

Re: E2 HTTP functions - Usage and good practices.

Excellent tutorial, I want to think of something to try them with now. Do we have any urlencode functions so we could send data to a server as part of the URL?

Share

Reply With Quote

09-03-2010

#6

Filipe

Mario, hard at work

MEMBER



Join Date: Jun 2007
Location: Portugal - Lisbon
Posts: 401
Blog Entries: 1

Re: E2 HTTP functions - Usage and good practices.

Updated with a section on `urlencode()`.

Share

beep boop

Reply With Quote

09-21-2010

#7

thetree

Wire Sofaking

MEMBER



Join Date: Oct 2009
Location: England
Posts: 952

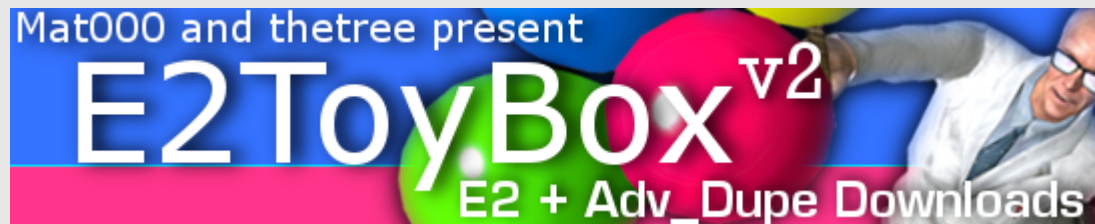
Re: E2 HTTP functions - Usage and good practices.

Wow i'm glad I found this, excellent way to understand HTTP functions as I couldnt quite understand how to send data.
Thanks alot

tree

Share

Gmessenger



Reply With Quote

« Previous Thread | Next Thread »

Bookmarks






Digg

Posting Permissions

You may not post new threads

BB code is On

 del.icio.us
 StumbleUpon
 Google
Facebook

You may not post replies
You may not post attachments
You may not edit your posts

Smilies are On
[IMG] code is On
[VIDEO] code is On
HTML code is Off
Trackbacks are On

Pingbacks are On
Refbacks are On

Forum Rules

-- Wiremod Reborn ▾

[Contact Us](#) [Wiremod.com](#) - Home of The Wiremod Addon [Archive](#) [Privacy Statement](#) [Top](#)

All times are GMT -7. The time now is 03:41 AM.

Powered by [vBulletin®](#) Version 4.2.0
Copyright © 2012 vBulletin Solutions, Inc. All rights reserved.
SEO by [vBSEO](#)