

Forum:[Hack Forums](#)

Thread: [Http Synchronous Class](#)

DBots - January 8, 2015, 8:34 am

I started a small project today which is a wrapper for the HttpWebRequest/HttpWebResponse synchronous methods.

I kept it very basic because I wanted it to be easy to use without having so many variables etc. so pretty much the guidelines I kept were:

1. Did the request go through fine?
 - If yes, work with the response.
- 2.If the request errored, was it a timeout?
 - If so, change proxy, etc, and reissue the request.
 - If not, check the exception or log it, or change proxy, etc, and reissue the request.

This is the code that I use to send and receive a request:

PHP Code

<?php

Dim clientProperties As New HttpProperties

With clientProperties

.Timeout = 100000

.ReadWriteTimeout = 100000

.MaximumRedirections = 7

.ProxyConnection = True

.Proxy = New WebProxy("127.0.0.1:8080")

.Url = New Uri("http://www.google.com")

.Accept = "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8"

.AcceptEncoding = "gzip, deflate, sdch"

.AcceptLanguage = "en-US,en;q=0.8"

.UserAgent =

"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36"

End With

Using response As HttpResponseMessage = HttpRequest.Get(clientProperties)

If Not response.Success Then

If response.TimedOut Then

Console.WriteLine("Time out message: " & response.Exception.Message)

Else

Console.WriteLine("Error: " & response.Exception.Message)

Console.WriteLine("Body:{0}{1}", Environment.NewLine, HttpRequest.ReadAsString(response))

End If

Else

Console.WriteLine("Headers:{0}{1}", Environment.NewLine, HttpRequest.ReadHeaders(response))

Dim str As String = HttpRequest.ReadAsString(response)

```

If str.Length = 0 Then
    Console.WriteLine("Empty Body..")
Else
    Console.WriteLine(str)
End If
End If
End Using

```

Now as you can see HttpRequest.Get/Post are shared functions which take in ClientProperties byref.

Is it bad practice to pass byref on a shared function? Is this even threadsafe?

Here is the code for the classes:

<http://pastebin.com/GwaysRUn>

idb. - January 8, 2015, 8:39 am

(Jan 8, 2015 09:34 AM)DBots Wrote: [\[CITE\]](#) I started a small project today which is a wrapper for the HttpRequest/HttpWebResponse synchronous methods.

I kept it very basic because I wanted it to be easy to use without having so many variables etc. so pretty much the guidelines I kept were:

1. Did the request go through fine?
 - If yes, work with the response.
- 2.If the request errored, was it a timeout?
 - If so, change proxy, etc, and reissue the request.
 - If not, check the exception or log it, or change proxy, etc, and reissue the request.

This is the code that I use to send and receive a request:

PHP Code

```
<?php
```

```
Dim clientProperties As New HttpProperties
```

```
With clientProperties
```

```
.Timeout = 100000
```

```
.ReadWriteTimeout = 100000
```

```
.MaximumRedirects = 7
```

```
.ProxyConnection = True
```

```
.Proxy = New WebProxy("127.0.0.1:8080")
```

```
.Url = New Uri("http://www.google.com")
```

```
.Accept = "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8"
```

```
.AcceptEncoding = "gzip, deflate, sdch"
```

```
.AcceptLanguage = "en-US,en;q=0.8"
```

```
.UserAgent =
```

```
"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36"
```

```
End With
```

```
Using response As HttpResponseMessage = HttpRequest.Get(clientProperties)
```

```
If Not response.Success Then
```

```
If response.TimedOut Then
```

```
Console.WriteLine("Time out message: " & response.Exception.Message)
```

```
Else
```

```

Console.WriteLine("Error: " & response.Exception.Message)
Console.WriteLine("Body:{0}{1}", Environment.NewLine, HttpRequest.ReadAsString(response))
End If
Else
Console.Write("Headers:{0}{1}", Environment.NewLine, HttpRequest.ReadHeaders(response))

Dim str As String = HttpRequest.ReadAsString(response)
If str.Length = 0 Then
Console.WriteLine("Empty Body..")
Else
Console.WriteLine(str)
End If
End If
End Using

```

Now as you can see HttpRequest.Get/Post are shared functions which take in ClientProperties byref.

Is it bad practice to pass byref on a shared function? Is this even threadsafe?

Here is the code for the classes:

<http://pastebin.com/GwaysRUn>

Some of this code seems strangely familiar. ?

Only3 - January 8, 2015, 8:49 am

(Jan 8, 2015 09:39 AM)idb. Wrote: [\[CITE\]](#) Some of this code seems strangely familiar. ?
? also this code seems shortly

DBots - January 8, 2015, 9:03 am

(Jan 8, 2015 09:39 AM)idb. Wrote: [\[CITE\]](#) Some of this code seems strangely familiar. ?
If you're insinuating I took code from your class and slapped it into mine then you're dead wrong mister. In that case I wouldn't post it here where I know you are a member and can easily call me out for it. I'd probably have gone asked in stackoverflow or something.

(Jan 8, 2015 09:49 AM)Only3 Wrote: [\[CITE\]](#) ? also this code seems shortly
Meaning..?

If you're being sarcastic then well the classes are not short but it would be a compiled dll which would make the usage code short. At least that was the idea.

Also, @idb, which part do you think I copied or what?

Forgot to say, I probably should have made HttpRequest MustInherit since it's unusable on it's own.

Anyone else have any USEFUL replies?

Only3 - January 8, 2015, 9:14 am

? it wasnt for u meaning its shoreter then idb

Tedd - January 8, 2015, 11:26 am

(Jan 8, 2015 10:14 AM)Only3 Wrote: [\[CITE\]](#) ? it wasnt for u meaning its shoreter then idb
This guy is like either 12 or foreign.

He means that your code is "shorter" than idb's, as in it's either less extensive or that you copied some of idb so obviously your code would be shorter than his.

DBots - January 9, 2015, 4:59 am

(Jan 8, 2015 12:26 PM)Tedd Wrote: [\[CITE\]](#) This guy is like either 12 or foreign.

He means that your code is "shorter" than idb's, as in it's either less extensive or that you copied some of idb so obviously your code would be shorter than his.

I guess the only way to know is to compare them side by side. My source is there so I have nothing to hide.

I actually coded this in terms of easy usage as i mentioned before. So believe it or not, I started with the code in terms of usage and then went on to the implementation which are those classes linked.

My original question though, was:

HttpRequest.Get/Post are shared functions which take in ClientProperties byref and make changes to it.

Is it bad practice to pass byref on a shared function? Is this even threadsafe?

ByKemal - June 10, 2016, 5:03 am

thanks, but do not working pastebin link?

Can you re-upload pastebin?

Ninja - June 10, 2016, 9:09 am

This just looks like a normal HttpWebRequest to me...

EDIT: Is this asynchronous or synchronous?