

Forum:[Hack Forums](#)

Thread: [Need Help With My Proxy Scraper](#)

Charcul - October 1, 2017, 6:44 pm

hello, i dont know a whole lot of c# and im currently making a proxy scraper

i can proxy scrape from sites such as proxy daily and blogspot proxies etc

But how would you scrape from sites like gatherproxy and hma, in which the ip and port are not combined by a :

Is this even possible?

Any help is greatly appreciated thanks

botland - October 1, 2017, 6:52 pm

htmlagilitypack is powerfull libary to lookup into html DOC

You can crate loop to scrap everything from website declaring right nodes in few seconds.

After that, you can test htmlresponse and check every IP and get pure quality proxies.

Charcul - October 1, 2017, 7:05 pm

(Oct 1, 2017 06:52 PM)botland Wrote: [\[CITE\]](#) htmlagilitypack is powerfull libary to lookup into html DOC

You can crate loop to scrap everything from website declaring right nodes in few seconds.

After that, you can test htmlresponse and check every IP and get pure quality proxies.

Thanks a lot mate will check this out :)

Omtism - October 2, 2017, 8:23 pm

(Oct 1, 2017 07:05 PM)Charcul Wrote: [\[CITE\]](#) Thanks a lot mate will check this out :)

You can also do it the autistic way and use the webbrowser control if you get stuck with agility.

goshmulticoloredalto - October 2, 2017, 9:27 pm

(Oct 1, 2017 07:05 PM)Charcul Wrote: [\[CITE\]](#) Thanks a lot mate will check this out :)

(Oct 2, 2017 08:23 PM)Omtism Wrote: [\[CITE\]](#) You can also do it the autistic way and use the webbrowser control if you get stuck with agility.

Why not just use HttpWebRequests?

They're really not that hard to figure out, hardest part will be regexing the proxies out of the response.

Have a look at this, should help you learn regex: <https://regexr.com/>

Lync - October 2, 2017, 10:21 pm

HttpRequest with Regex will be the fastest and most stable method by far. You don't need all that overhead with a 3rd party library or the nasty .NET WebBrowser control.

GatherProxy: Stores their proxies in JSON format, which makes extracting this info almost as easy as IP:PORT string formatted proxies.

You can use Newtonsofts Json.net library with Linq and extract the data via JObject/JArray/JToken. If you want to use something from .NET go with System.Web.Script.Serialization

HideMyAss: last I checked they have temporarily suspended their free proxy list service.

vercenta - October 3, 2017, 5:07 am

Each site can have different html so the ip:port regex method only work sometimes.

Otherwise you have to parse through the html tags.

btw regex is inefficient af

Ninja - October 3, 2017, 4:37 pm

To anyone suggesting Regex for this; get out.

Use WebRequests to download the page contents, and HtmlAgilityPack or AngleSharp to parse the proxies.

Dianab0lical - October 4, 2017, 8:03 am

(Oct 3, 2017 04:37 PM)Ninja Wrote: [\[CITE\]](#) To anyone suggesting Regex for this; get out.

Use WebRequests to download the page contents, and HtmlAgilityPack or AngleSharp to parse the proxies.

I've always heard Regex is extremely inefficient, so I've tried to avoid it except in situations where Regex directly fits what I need to accomplish.

My question to you is why is that the consensus? What does Regex do that makes it such a problem? And how do frameworks like the HTMLAgilityPack handle it differently?

Owl - October 4, 2017, 8:49 am

(Oct 4, 2017 08:03 AM)H4rdBeat Wrote: [\[CITE\]](#) I've always heard Regex is extremely inefficient, so I've tried to avoid it except in situations where Regex directly fits what I need to accomplish.

My question to you is why is that the consensus? What does Regex do that makes it such a problem? And how do frameworks like the HTMLAgilityPack handle it differently?

It's difficult to correctly parse through html. Inefficient patterns may cause excessive backtracking (quite common with people who blindly c+p patterns) which will be noticeable on heavy/repeated processing.

Dianab0lical - October 5, 2017, 6:36 am

(Oct 4, 2017 08:49 AM)Orka Wrote: [\[CITE\]](#) It's difficult to correctly parse through html. Inefficient patterns may cause excessive backtracking (quite common with people who blindly c+p patterns) which will be noticeable on heavy/repeated processing.

So I guess my follow-up to this is: with proper and precise matching patterns, on data you're only going to parse one time, is it really that detrimental?

And also, can you give me a nutshell of how the HTMLAgilityPack handles parsing (if you wouldn't mind, just a short/sweet one-liner)?

Owl - October 5, 2017, 8:31 am

(Oct 5, 2017 06:36 AM)H4rdBeat Wrote: [\[CITE\]](#) So I guess my follow-up to this is: with proper and precise matching patterns, on data you're only going to parse one time, is it really that detrimental?

And also, can you give me a nutshell of how the HTMLAgilityPack handles parsing (if you wouldn't mind, just a short/sweet one-liner)?

It again comes down to the input and what you need from it, you will still have to start the regex engine.

With hap, you select nodes using xpath expressions. Unlike regex, parsers are aware of html and how poorly it can be written.

Dianab0lical - October 5, 2017, 8:49 am

(Oct 5, 2017 08:31 AM)Orka Wrote: [\[CITE\]](#) It again comes down to the input and what you need from it, you will still have to start the regex engine.

With hap, you select nodes using xpath expressions. Unlike regex, parsers are aware of html and how poorly it can be written.

That makes a lot more sense. Appreciate the response man! Cheers.

Ninja - October 5, 2017, 12:32 pm

(Oct 5, 2017 08:31 AM)Orka Wrote: [\[CITE\]](#) It again comes down to the input and what you need from it, you will still have to start the regex engine.

With hap, you select nodes using xpath expressions. Unlike regex, parsers are aware of html and how poorly it can be written.

Fuck outta here with that bullshit. The grasshopper algorithm is superior to any form of parsing.

fxbgghacker - October 6, 2017, 8:06 am

(Oct 2, 2017 10:21 PM)Pitri Wrote: [\[CITE\]](#) HttpWebRequest with Regex will be the fastest and most stable method by far. You don't need all that overhead with a 3rd party library or the nasty .NET WebBrowser control.

GatherProxy: Stores their proxies in JSON format, which makes extracting this info almost as easy as IP:PORT string formatted proxies.

You can use Newtonsofts Json.net library with Linq and extract the data via JObject/JArray/JToken. If you want to use something from .NET go with System.Web.Script.Serialization

HideMyAss: last I checked they have temporarily suspended their free proxy list service. I second this method. (pretty much how all my scrapers work lol)

M O D E R N - October 6, 2017, 8:08 am

If you want you can take some stuff from my crappy checker:

<https://hackforums.net/showthread.php?tid=5730642>

Ninja - October 6, 2017, 8:49 am

(Oct 2, 2017 10:21 PM)Pitri Wrote: [\[CITE\]](#) Regex will be the fastest and most stable method by far. You don't need all that overhead with a 3rd party library

I can't even. You clearly don't know what you are talking about. In most cases, a HTML parsing library is more efficient than Regex. Why? Because its a library made for HTML parsing. And what overhead are you talking about? Do you even know what overhead is? Or are you just spewing out random words.