

# Forum:[Hack Forums](#)

Thread: [Random User-Agent Generator](#)

glitchgxd - July 26, 2017, 1:11 pm

Was making this for a project I'm working on and thought maybe someone might want something like this. It's pretty simple but efficient if you're trying to make multiple automated requests.

Just include FakeAgent.cs in your project (and change the namespaces of course), and add agent\_list.json as a new file resource to your project (Encoding UTF-8, FileType Text).

FYI: This requires the NuGet package for Newtonsoft.Json so that will need to be included in your project unless you want to change/customize the de-serialization method!

FakeAgent.cs: <https://pastebin.com/fsSzTGf2> | (Raw) <https://pastebin.com/raw/fsSzTGf2>  
agent\_list.json: <https://pastebin.com/NRvZQ1Bj> | (Raw) <https://pastebin.com/raw/NRvZQ1Bj>

Usage:

```
Code FakeAgent agent = new FakeAgent();
for (int i = 0; i < 10; i++)
    WriteLog("Random agent: {0}", agent.New.Random().AgentString);
for (int i = 0; i < 10; i++)
    WriteLog("Random iOS agent: {0}", agent.New.OSType("iOS").Random().AgentString);
for (int i = 0; i < 10; i++)
    WriteLog("Random Console agent: {0}",
agent.New.DeviceType("Console").Random().AgentString);
for (int i = 0; i < 10; i++)
    WriteLog("Random browser/windows/desktop agent: {0}",
agent.New.AgentType("Browser").OSType("Windows").DeviceType("Desktop").Random().AgentString);
for (int i = 0; i < 10; i++)
    WriteLog("Random crawler agent: {0}", agent.New.AgentType("Crawler").Random().AgentString);
Output:
```

Code Random agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36  
Random agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_10\_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.2357.124 Safari/537.36  
Random agent: Mozilla/2.0 (compatible; Ask Jeeves/Teoma)  
Random agent: Mozilla/5.0 (iPhone; CPU iPhone OS 8\_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) CriOS/43.0.2357.61 Mobile/12F70 Safari/600.1.4  
Random agent: Mozilla/5.0 (Android; Tablet; rv:44.0) Gecko/44.0 Firefox/44.0  
Random agent: Mozilla/5.0 (PLAYSTATION 3; 2.00)  
Random agent: Mozilla/5.0 (PLAYSTATION 3; 1.70)  
Random agent: Mozilla/5.0 (PlayStation Vita 3.50) AppleWebKit/537.73 (KHTML, like Gecko) Silk/3.2  
Random agent: Mozilla/5.0 (iPad; CPU OS 8\_2 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12D508 Safari/600.1.4

Random agent: Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/538.15 (KHTML, like Gecko) Version/8.0 Safari/538.15 Ubuntu/14.10 (3.10.3-0ubuntu3) Epiphany/3.10.3

Random iOS agent: Mozilla/5.0 (iPad; CPU OS 8\_1\_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12B466 Safari/600.1.4

Random iOS agent: Mozilla/5.0 (iPad; CPU OS 8\_4 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12H143 Safari/600.1.4

Random iOS agent: Mozilla/5.0 (iPad; CPU OS 8\_1\_1 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12B436 Safari/600.1.4

Random iOS agent: Mozilla/5.0 (iPad; CPU OS 8\_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12F69 Safari/600.1.4

Random iOS agent: Mozilla/5.0 (iPhone; CPU iPhone OS 8\_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) CriOS/42.0.2311.47 Mobile/12F70 Safari/600.1.4

Random iOS agent: Mozilla/5.0 (iPad; CPU OS 8\_4 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12H143 Safari/600.1.4

Random iOS agent: Mozilla/5.0 (iPad; CPU OS 8\_1\_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12B466 Safari/600.1.4

Random iOS agent: Mozilla/5.0 (iPhone; CPU iPhone OS 8\_1 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12B411 Safari/600.1.4

Random iOS agent: Mozilla/5.0 (iPad; CPU OS 8\_4 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12H143 Safari/600.1.4

Random iOS agent: Mozilla/5.0 (iPhone; CPU iPhone OS 8\_1 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12B411 Safari/600.1.4

Random Console agent: Opera/9.30 (Nintendo Wii; U; ; 2047-7; en)

Random Console agent: Mozilla/5.0 (PLAYSTATION 3; 1.0)

Random Console agent: Mozilla/5.0 (PLAYSTATION 3; 4.60) AppleWebKit/531.22.8 (KHTML, like Gecko)

Random Console agent: Mozilla/5.0 (PlayStation 4 2.03) AppleWebKit/537.73 (KHTML, like Gecko)

Random Console agent: Mozilla/5.0 (PLAYSTATION 3; 4.50) AppleWebKit/531.22.8 (KHTML, like Gecko)

Random Console agent: Mozilla/5.0 (PlayStation 4 2.04) AppleWebKit/537.73 (KHTML, like Gecko)

Random Console agent: Mozilla/5.0 (PlayStation 4 2.00) AppleWebKit/537.73 (KHTML, like Gecko)

Random Console agent: Opera/9.30 (Nintendo Wii; U; ; 2047-7; en)

Random Console agent: Mozilla/5.0 (PLAYSTATION 3; 3.55)

Random Console agent: Mozilla/5.0 (PlayStation Vita 3.36) AppleWebKit/537.73 (KHTML, like Gecko) Silk/3.2

Random browser/windows/desktop agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.152 Safari/537.36

Random browser/windows/desktop agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.124 Safari/537.36

Random browser/windows/desktop agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.0

Random browser/windows/desktop agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.2357.125 Safari/537.36 OPR/30.0.1835.88

Random browser/windows/desktop agent: Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.52 Safari/537.36 OPR/31.0.1889.50 (Edition beta)

Random browser/windows/desktop agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/6.0)

Random browser/windows/desktop agent: Mozilla/5.0 (Windows NT 6.1; rv:36.0) Gecko/20100101 Firefox/36.0 Seamonkey/2.33.1

Random browser/windows/desktop agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.130 Safari/537.36

Random browser/windows/desktop agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.81 Safari/537.36 OPR/30.1835.49

(Edition beta)

Random browser/windows/desktop agent: Mozilla/5.0 (compatible; MSIE 10.6; Windows NT 6.1; Trident/5.0; InfoPath.2; SLCC1; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR 2.0.50727) 3gpp-gba UNTRUSTED/1.0

Random crawler agent: Mozilla/5.0 (compatible; bingbot/2.0 +http://www.bing.com/bingbot.htm)

Random crawler agent: YahooSeeker/1.2 (compatible; Mozilla 4.0; MSIE 5.5; yahooseeker at yahoo-inc dot com ; http://help.yahoo.com/help/us/shop/merchant/)

Random crawler agent: BaiDuSpider

Random crawler agent: Googlebot/2.1 (+http://www.google.com/bot.html)

Random crawler agent: Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)

Random crawler agent: Baiduspider+(+http://www.baidu.com/search/spider\_jp.html)

Random crawler agent: Baiduspider+(+http://www.baidu.com/search/spider\_jp.html)

Random crawler agent: Mozilla/5.0 (compatible; Yahoo! Slurp China; http://misc.yahoo.com.cn/help.html)

Random crawler agent: Googlebot/2.1 (+http://www.google.com/bot.html)

Random crawler agent: BaiDuSpider

Credits to the guy on github who made a user agent list generator in PHP. I just took the agent\_list and modified it for C# use:

[https://github.com/joecampo/random-user-agent/blob/master/src/agents/agent\\_list.json](https://github.com/joecampo/random-user-agent/blob/master/src/agents/agent_list.json)

If you have any questions / suggestions, feel free to let me know. But please do a little research first if it's something basic like adding resources or new classes to a project.

**Ninja** - July 26, 2017, 1:59 pm

The property in the fake agent class confuses me. First you create an array, then copy a list to that array, then convert the array to a list. Why?

**SL x TnT** - July 26, 2017, 3:27 pm

(Jul 26, 2017 01:59 PM)Ninja Wrote: [\[CITE\]](#) The property in the fake agent class confuses me. First you create an array, then copy a list to that array, then convert the array to a list. Why?

I think his end goal was to have a copy of the list of agents to prevent InvalidOperationException exceptions on the filtering/random methods. Returning the ToList is enough.

The Skip in the Random method is weird to me. Easier to return the item at the index instead of skipping. It's also not thread safe.

**glitchgxd** - July 26, 2017, 3:28 pm

(Jul 26, 2017 01:59 PM)Ninja Wrote: [\[CITE\]](#) The property in the fake agent class confuses me. First you create an array, then copy a list to that array, then convert the array to a list. Why?

I was messing around with it a lot and was modifying the collection at one point. Now that it's not being modified the copy doesn't really need to be there.

(Jul 26, 2017 03:27 PM)SL x TnT Wrote: [\[CITE\]](#) I think his end goal was to have a copy of the list of agents to prevent InvalidOperationException exceptions on the filtering/random methods. Returning the ToList is enough.

The Skip in the Random method is weird to me. Easier to return the item at the index instead of skipping. It's also not thread safe.

Good point, should just directly access agentList[random.Next(agentList.Count())]

**SL x TnT** - July 26, 2017, 3:38 pm

(Jul 26, 2017 03:28 PM)Gl?tch Wrote: [\[CITE\]](#) I was messing around with it a lot and was modifying the collection at one point. Now that it's not being modified the copy doesn't really need to be there.

Good point, should just directly access `agentList[random.Next(agentList.Count())]`  
Don't forget about making it thread safe. If you remove that "New" property, all the other methods aren't thread safe either.

**glitchgxd** - July 26, 2017, 3:43 pm

(Jul 26, 2017 03:38 PM)SL x TnT Wrote: [\[CITE\]](#) Don't forget about making it thread safe. If you remove that "New" property, all the other methods aren't thread safe either.

You mentioned only needing to use `ToList` instead of the array copy -- does `ToList()` return a new object unlinked from the original list? I find myself getting hooked on references a lot when dealing with `IEnumerables/Lists`.

**SL x TnT** - July 26, 2017, 4:56 pm

(Jul 26, 2017 03:43 PM)GI?tch Wrote: [\[CITE\]](#) You mentioned only needing to use `ToList` instead of the array copy -- does `ToList()` return a new object unlinked from the original list? I find myself getting hooked on references a lot when dealing with `IEnumerables/Lists`.

The List will be a new object, but the items inside will still have the references to the old objects.

**0xDEAD10CC** - July 26, 2017, 7:55 pm

This isn't very well designed...

To solve that references issue - you could create a new list with the capacity of the count of the first (prevent reallocation), and return the new `List<T>` to return. Add them to the list, no need for an array. Create your own `Clone()` method (not implemented from `ICloneable`), and return the type rather than object as another more abstract solution that lets the implementation define how the copy should take place and whether it's a shallow or deep copy.

Code public interface `ICloneable<out T>`

```
{  
    T Clone();  
}
```

private class `Obj : ICloneable<Obj>`

```
{  
    public int Value { get; set; }  
  
    public Obj(int value)  
    {  
        Value = value;  
    }  
  
    public Obj Clone()  
    {  
        return new Obj(Value);  
    }  
  
    public override string ToString()  
    {  
        return $"{Value}";  
    }  
}
```

```

public static void Main(string[] args)
{
    List<Obj> list1 = new List<Obj>
    {
        new Obj(1),
        new Obj(2),
        new Obj(3)
    };

    List<Obj> list2 = new List<Obj>(list1.Count);
    list1.ForEach(o => list2.Add(o.Clone()));
    list2[0].Value = 9;

    list1.ForEach(Console.WriteLine);
    Console.WriteLine();
    list2.ForEach(Console.WriteLine);
}

```

The argument against ICloneable is that there's no description of whether it provides a deep or shallow copy of the original, yet sometimes it's useful to have a clone regardless IMO, just not sure why they haven't depreciated the first yet after so long and also haven't come out with a covariant type generic ICloneable if they've kept the original interface around for so long...

**J\_128** - August 13, 2017, 12:43 am

Great share thanks you bro ;)

**Milan Hermanus** - August 13, 2017, 2:59 am

FakeAgent.cs.61 - use String.Compare() instead of calling tolower() on both strings

why are you skipping instead of accessing directly via .At() or the indexer?

**0xDEAD10CC** - August 13, 2017, 11:50 am

(Aug 13, 2017 02:59 AM)Milan Hermanus Wrote: [\[CITE\]](#) FakeAgent.cs.61 - use String.Compare() instead of calling tolower() on both strings

why are you skipping instead of accessing directly via .At() or the indexer?  
What's wrong with string.Equals() for equality?

**glitchgxd** - August 13, 2017, 1:45 pm

(Aug 13, 2017 02:59 AM)Milan Hermanus Wrote: [\[CITE\]](#) FakeAgent.cs.61 - use String.Compare() instead of calling tolower() on both strings

why are you skipping instead of accessing directly via .At() or the indexer?  
Never knew about or used .At() for some reason. Will remember that for the future.

I've ran into weird issues in the past using comparison functions, so I tend to avoid them. Also, the tolower() == tolower() is more easily read for me, and I just prefer that way of comparing.

**0xDEAD10CC** - August 13, 2017, 3:07 pm

(Aug 13, 2017 01:45 PM)Gl?tch Wrote: [\[CITE\]](#) Never knew about or used .At() for some reason. Will remember that for the future.

I've ran into weird issues in the past using comparison functions, so I tend to avoid them. Also, the tolower() == tolower() is more easily read for me, and I just prefer that way of comparing.

It's not about what is easier, it's about accuracy. When you start dealing with locale you're going to have major issues comparing by using ToLower() or ToUpper() before a direct comparison using the == operator. Equals has a StringComparison parameter that you should be using.

**gigajew** - August 14, 2017, 10:30 am

Here's how I would do it. Doesn't require internet connection and doesn't contain a huge list or a JSON parser

#### Sample output

Code Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_12\_4) Gecko/20100101 (KHTML, like Gecko)  
Chrome/86.0.2910.119 Safari/430.30  
Mozilla/5.0 (X11; Linux x86\_64; rv:57.7) Gecko/20100101 (KHTML, like Gecko) Version/9.0  
Mobile/14G60 Safari/235.6  
Mozilla/5.0 (X11; Linux x86\_64; rv:0.9) Gecko/20100101 (KHTML, like Gecko)  
Chrome/49.0.2885.189 Safari/415.48  
Mozilla/5.0 (Windows NT 6.9; WOW64; rv:81.9) Gecko/20100101 (KHTML, like Gecko) Version/4.0  
Mobile/14G60 Safari/216  
Mozilla/5.0 (Windows NT 0.6; Win64; x64) Gecko/20100101 Firefox/58.0  
Mozilla/5.0 (X11; Linux x86\_64; rv:41.6) AppleWebKit/693.4.6.9 Firefox/57.0  
Mozilla/5.0 (Windows NT 5.4; WOW64; rv:81.4) AppleWebKit/949.7.4.5 Firefox/41.0  
Mozilla/5.0 (Windows NT 10.8; Win64; x64) AppleWebKit/349.2.2 (KHTML, like Gecko)  
Chrome/31.0.2331.178 Safari/543.34  
Mozilla/5.0 (compatible; MSIE 1.1; Windows NT 11; Trident/0; Trident/5.7) Gecko/20100101  
(KHTML, like Gecko) Chrome/55.0.1107.170 Safari/540.26  
Mozilla/5.0 (X11; Linux x86\_64; rv:72.4) AppleWebKit/411.9.8.6 (KHTML, like Gecko)  
Chrome/87.0.2081.102 Safari/576.31  
Mozilla/5.0 (Windows NT 5.1; WOW64; rv:28.3) AppleWebKit/691.5.1.3 (KHTML, like Gecko)  
Version/3.0 Mobile/14G60 Safari/543.7  
Mozilla/5.0 (compatible; MSIE 11.4; Windows NT 6.7; Trident/0.4; Trident/1) AppleWebKit/952.6.2.9  
(KHTML, like Gecko) Version/8.0 Mobile/14G60 Safari/509.2  
Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_12\_6) AppleWebKit/192.3.4.5 (KHTML, like Gecko)  
Chrome/33.0.1076.162 Safari/530.49  
Mozilla/5.0 (iPad; CPU OS 10\_2\_2 like Mac OS X) Gecko/20100101 Firefox/58.0  
Mozilla/5.0 (X11; Linux x86\_64; rv:80.3) AppleWebKit/465.6.3.5 (KHTML, like Gecko)  
Chrome/46.0.2179.183 Safari/591.39  
Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_11\_1) AppleWebKit/239.6.8.9 (KHTML, like Gecko)  
Version/5.0 Mobile/14G60 Safari/583.6  
Mozilla/5.0 (compatible; MSIE 9.6; Windows NT 3.9; Trident/6.3; Trident/0.5) Gecko/20100101  
Firefox/55.0  
Mozilla/5.0 (Windows NT 3.8; Win64; x64) Gecko/20100101 (KHTML, like Gecko) Version/4.0  
Mobile/14G60 Safari/532.1  
Mozilla/5.0 (Windows NT 7.3; Win64; x64) Gecko/20100101 (KHTML, like Gecko)  
Chrome/70.0.1463.108 Safari/596.28  
Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:33.7) AppleWebKit/184.6.7.1 (KHTML, like Gecko)  
Chrome/33.0.1280.199 Safari/428.30

**Milan Hermanus** - August 14, 2017, 11:02 am

(Aug 13, 2017 11:50 AM)0xDEAD10CC Wrote: [CITE](#) What's wrong with string.Equals() for equality?  
excuse me, i meant that one.

**0xDEAD10CC** - August 14, 2017, 11:41 am

(Aug 14, 2017 10:30 AM)gigajew Wrote: [\[CITE\]](#) Here's how I would do it. Doesn't require internet connection and doesn't contain a huge list or a JSON parser

...

Ever used switch statements before?.. You might as well make the class static since there's nothing that really requires an instance of the class in there at this point. The structure of most of the code could be improved too. There's some obvious redundancy in the flow of your logic.

ie.

Code private static string GetRandomArchitecture(System system)

```
{
    var i = _random.Next(0, 2);
    if (system == System.Windows)
    {
        return (i == 1) ? "Win64; x64" : $"WOW64; rv:{GetRandomVersion(80)}";
    }
    return string.Empty;
}
```

If this function only returns something meaningful when the system is Windows, then why calculate a random value outside of the if statement when it's not required to return string.Empty?

Personally I think a configuration file still provides more control around User Agents however, and that's why I would go that route still. What if you want a completely custom user agent that could imitate a Google Bot for instance? Now you would have to recompile your code to include that...

**Ninja** - August 14, 2017, 9:32 pm

(Aug 14, 2017 11:41 AM)0xDEAD10CC Wrote: [\[CITE\]](#) Ever used switch statements before?.. You might as well make the class static since there's nothing that really requires an instance of the class in there at this point.

Even so, if you wanna follow OOP strict you shouldn't make it static still.

**0xDEAD10CC** - August 14, 2017, 11:16 pm

(Aug 14, 2017 09:32 PM)Ninja Wrote: [\[CITE\]](#) Even so, if you wanna follow OOP strict you shouldn't make it static still.

Why would you need OOP here? It's a generator for user agent strings, there isn't much you can derive from this. As far as I see this, it's just a collection of functions that do 1 thing. Any abstract layer on top of this adds an unnecessary window of complexity around this functionality IMHO..

If this was going into a larger project like some web bot, maybe each bot would implement their own custom User Agent generator, in which case it might make sense, but without that context it's completely pointless to think about OOP at this time.

**Milan Hermanus** - August 15, 2017, 10:02 am

(Aug 14, 2017 09:32 PM)Ninja Wrote: [\[CITE\]](#) Even so, if you wanna follow OOP strict you shouldn't make it static still.

You're not supposed to blindly follow OOP.

You use OOP when necessary