

Forum:[Hack Forums](#)

Thread: [Adding text to .txt file in webserver](#)

Formula 1 - April 23, 2020, 7:15 am

Hello,

I made a logs.txt to see how many times user connects to my application, i saved it in .txt file (On my pc)

But i want to do the same thing to a file (log.txt) on webserver

Code /*

```
FileStream stream = null;
string fileName = @"E:/logs.txt";
stream = new FileStream(fileName, FileMode.OpenOrCreate);
using (StreamWriter writer = new StreamWriter(stream, Encoding.UTF8)) ;
try
{
    using (var writer = File.AppendText(fileName))
    {
        writer.WriteLine(welcome.Text);
    }
}
catch (Exception exp)
{
    Console.Write(exp.Message);
}
*/
```

gigajew - April 23, 2020, 8:12 am

What kind of server side software are you running?

FTP? Php?

Shadow_Cloud - April 23, 2020, 8:14 am

As per another post

I would strongly advise log4net for this. What you are doing here isn't thread safe and could easily result in exceptions.

Code class Program

```
{
    private static readonly log4net.ILog log =
log4net.LogManager.GetLogger(System.Reflection.MethodBase.GetCurrentMethod().DeclaringType
);

    static void Main(string[] args)
    {
```

```

    log.Info("Hello logging world!");
    Console.WriteLine("Hit enter");
    Console.ReadLine();
}
}

```

Have a look here > <https://stackify.com/log4net-guide-dotnet-logging/>

Formula 1 - April 23, 2020, 9:29 am

(Apr 23, 2020 08:12 AM)gigajew Wrote: [\[CITE\]](#) What kind of server side software are you running?

FTP? Php?
FTP brother.

gigajew - April 23, 2020, 10:08 am

(Apr 23, 2020 09:29 AM)uberhacker Wrote: [\[CITE\]](#) FTP brother.

Here you go

<https://docs.microsoft.com/en-us/dotnet/framework/network-programming/how-to-upload-files-with-ftp>

notahackerman - April 23, 2020, 10:20 am

(Apr 23, 2020 07:15 AM)uberhacker Wrote: [\[CITE\]](#) Hello,

I made a logs.txt to see how many times user connects to my application, i saved it in .txt file (On my pc)

But i want to do the same thing to a file (log.txt) on webserver

Code /*

```

    FileStream stream = null;
    string fileName = @"E:/logs.txt";
    stream = new FileStream(fileName, FileMode.OpenOrCreate);
    using (StreamWriter writer = new StreamWriter(stream, Encoding.UTF8)) ;
    try
    {
        using (var writer = File.AppendText(fileName))
        {
            writer.WriteLine(welcome.Text);
        }
    }
    catch (Exception exp)
    {
        Console.Write(exp.Message);
    }
}
*/

```

pm me: hackerman#0001

????? - April 23, 2020, 8:01 pm

you're going to want to download the existing logs.txt from your webserver, append the new line and re-upload it making sure to overwrite the existing file.

If you don't want to hardcode your FTP creds into your application you should look into something like this

<https://www.php.net/manual/en/function.file-put-contents.php>

Where your application will make a single web request and the server will handle logging and writing to the file.

This is safer and will also help if your application has lots of users.

Imk if you need any help with this

Shadow Cloud - April 24, 2020, 2:50 am

(Apr 23, 2020 08:01 PM)????? Wrote: [\[CITE\]](#) you're going to want to download the existing logs.txt from your webserver, append the new line and re-upload it making sure to overwrite the existing file.

If you don't want to hardcode your FTP creds into your application you should look into something like this

<https://www.php.net/manual/en/function.file-put-contents.php>

Where your application will make a single web request and the server will handle logging and writing to the file.

This is safer and will also help if your application has lots of users.

Imk if you need any help with this

I wouldn't take the first approach, as this file grows it will become more and more bandwidth intensive. If you get to the point where that file is 2MB, you are downloading it, appending on the client, then uploading the 2MB again. And this will just get worse, plus the app has the details to connect which is already a code smell.

The second option is much better, but it's still important to make sure that when the server writes to disk its doing so in a thread safe manner.