

Forum:[Hack Forums](#)

Thread: [Am i correctly disposing this?](#)

kevy42 - August 17, 2017, 2:37 pm

Basicly, my program needs to call this method alot, and i was wondering if everytime it calls the method, the old instances(var client = new SmtpClient("smtp.gmail.com", 587) as an example) gets disposed, cashed in memory or overwritten. What im trying to say is that, is this the right way of doing it?

Code // Email sender method

```
public static void SendEmail(string To, string Title, string Body)
{
    var client = new SmtpClient("smtp.gmail.com", 587)
    {
        Credentials = new NetworkCredential(Username, Password),
        EnableSsl = true
    };

    client.Send(To, To, Title, Body);
}
```

+Haze - August 17, 2017, 2:54 pm

Well, if the SmtpClient instance needs to be only defined once then do so instead of creating multiple instances with the same credentials. Define the stuff that stays the same as the class level. Also, to effectively dispose of an object you'd need to use the using block.

Code using (var client = new SmtpClient(string, port)

```
{

}
```

Electron - August 17, 2017, 3:13 pm

It should be fine so since SmtpClient doesn't have a dispose method

Edit: Nvm, it actually implements IDisposable in .NET 4.0

0xDEAD10CC - August 17, 2017, 3:41 pm

Create a class that implements IDisposable to dispose of the SmtpClient internally once your class is done using it. You only need a single instance of the client, and since you'll be calling it many times constructing one with the same parameters it looks like, it would make more sense to only create it once, send a bunch of emails using the single instance, then dispose of it once you're done using it.

Btw - you're not at all properly disposing anything. You're just creating a bunch of instances it looks like, and the only thing that would cause the disposing would be the GC.

kevy42 - August 17, 2017, 3:47 pm

(Aug 17, 2017 02:54 PM)+Haze Wrote: [CITE](#) Well, if the SmtpClient instance needs to be only

defined once then do so instead of creating multiple instances with the same credentials. Define the stuff that stays the same as the class level. Also, to effectively dispose of an object you'd need to use the using block. Code using (var client = new SmtpClient(string, port) { }

(Aug 17, 2017 03:13 PM)1912 Wrote: [\[CITE\]](#) It should be fine so since SmtpClient doesn't have a dispose method Edit: Nvm, it actually implements IDisposable in .NET 4.0

(Aug 17, 2017 03:41 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Create a class that implements IDisposable to dispose of the SmtpClient internally once your class is done using it. You only need a single instance of the client, and since you'll be calling it many times constructing one with the same parameters it looks like, it would make more sense to only create it once, send a bunch of emails using the single instance, then dispose of it once you're done using it. Btw - you're not at all properly disposing anything. You're just creating a bunch of instances it looks like, and the only thing that would cause the disposing would be the GC.

Thank you all for the help, i solved the issue ?