The results are in! See what nearly 90,000 developers picked as their most loved, dreaded, and desired coding languages and more in the 2019 Developer Survey.

Sending email in .NET through Gmail

<div style="float:right">Ask Question</div>

▲

805

▼

★

349

Instead of relying on my host to send email, I was thinking of sending the email messages using my Gmail account. The emails are personalized emails to the bands I play on my show. Is it possible to do?

| c# | .net | email | smtp | gmail |

edited Nov 16 '17 at 1:50

Sameer
2,110   1   19   41

asked Aug 28 '08 at 13:28

Mike Wills
10.6k   26   77   138

<systemnetmail.com> is probably the most absurdly complete site dedicated to a *single* .NET namespace...but it has EVERYTHING you could ever want to know about sending mail via .NET, be it ASP.NET or Desktop. <*systemwebmail.com*> *was the original URL in the post, but should not be used for .NET 2.0 and above.* – Adam Haile Aug 28 '08 at 13:36 ✎

13   If you're using ASP.Net Mvc I would recommend having a look at MvcMailer: github.com/smsohan/MvcMailer/wiki/MvcMailer-Step-by-Step-Guide – noocyte Oct 5 '11 at 7:42

I have a nice example of how to do this on my site jarloo.com/gmail-in-c – Kelly Oct 14 '11 at 17:27

how to send file using this code ? – Ranadheer Reddy May 28 '12 at 12:24

possible duplicate of Sending email through Gmail SMTP server with C# – George Duckett Jun 10 '13 at 13:50

21 Answers

▲

1006

▼

✔

Be sure to use `System.Net.Mail`, not the deprecated `System.Web.Mail`. Doing SSL with `System.Web.Mail` is a gross mess of hacky extensions.

```
using System.Net;
using System.Net.Mail;

var fromAddress = new MailAddress("from@gmail.com", "From Name");
var toAddress = new MailAddress("to@example.com", "To Name");
const string fromPassword = "fromPassword";
const string subject = "Subject";
const string body = "Body";

var smtp = new SmtpClient
{
    Host = "smtp.gmail.com",
    Port = 587,
    EnableSsl = true,
    DeliveryMethod = SmtpDeliveryMethod.Network,
    UseDefaultCredentials = false,
    Credentials = new NetworkCredential(fromAddress.Address, fromPa
};
using (var message = new MailMessage(fromAddress, toAddress)
{
    Subject = subject,
    Body = body
})
{
    smtp.Send(message);
}
```

edited Aug 16 '14 at 9:49

Sam

5,535   8   36   61

answered Aug 28 '08 at 14:08

Domenic

80.1k   30   185   242

---

46   You can still get user not logged in errors if Google just suddenly decides you have sent too many in the past xx number of minutes. You should always add a trySend, if it errors sleep a while, and then attempt again. – Jason Short Aug 26 '09 at 6:30

69   Interesting note: If you swap 'UseDefaultCredentials = false,' and 'Credentials = ...' it won't authenticate. – Nathan Wheeler Nov 17 '09 at 16:56

11   There are no problems with SPF using this method. Every email client can be configured to do exactly this. You just may get problems if you use your own server (i.e. something else than `smtp.gmail.com`) with `something@gmail.com` as sender. Btw: `smtp.gmail.com` automatically overwrites the sender address if it's not yours. – Meinersbur Mar 18 '10 at 18:39

22   I was having a hard time getting this working even with trying various tweaks. As suggested on a related post, I found that it was actually my antivirus that was preventing emails from being successfully sent. The antivirus in question is McAffee, and its "Access Protection" has a "Antivirus Standard Protection" category that has a "Prevent mass mailing worms from sending email" rule. Tweaking / disabling that rule got this code working for me! – yourbuddypal Aug 5 '12 at 17:02

15   I was getting the 5.5.1 Authentication Required error message until I realized I was testing with an account (my personal one) that had two factor authentication turned on. Once I used an account that didn't have that, it worked fine. I could also have generated a password for my application that I was testing from in my personal acct, but I didn't want to do that. – Nick DeVore Jun 17 '13 at 16:18

Copying from another answer, the above methods work but gmail

always replaces the "from" and "reply to" email with the actual
sending gmail account. apparently there is a work around however:

http://karmic-development.blogspot.in/2013/10/send-email-from-
aspnet-using-gmail-as.html

"3. In the Accounts Tab, Click on the link "Add another email address
you own" then verify it"

Or possibly this

Update 3: Reader Derek Bennett says, "The solution is to go into
your gmail Settings:Accounts and "Make default" an account other
than your gmail account. This will cause gmail to re-write the From
field with whatever the default account's email address is."

answered Mar 22 '18 at 13:07

rogerdpack
35.4k　17　137　258

For me to get it to work, i had to enable my gmail account making it
possible for other apps to gain access. This is done with the "enable
less secure apps" and **also** using this link:
https://accounts.google.com/b/0/DisplayUnlockCaptcha

edited Mar 22 '18 at 13:04

rogerdpack
35.4k　17　137　258

answered Jan 3 '14 at 15:25
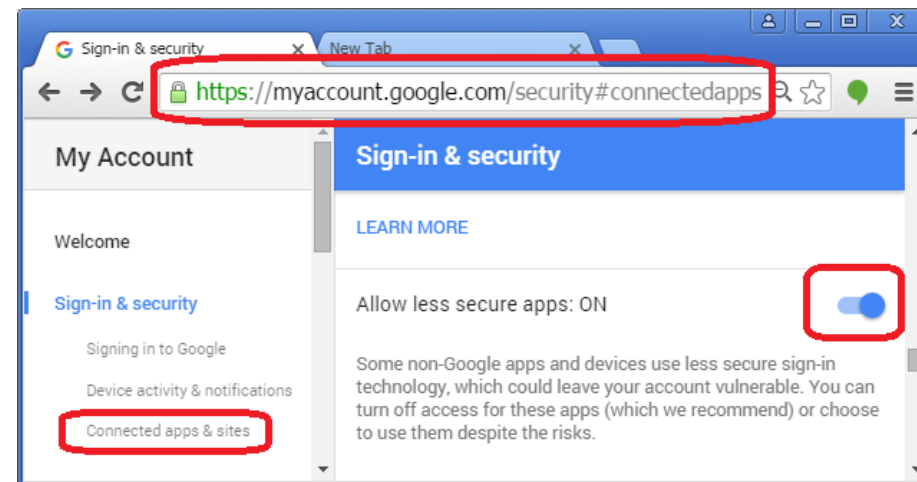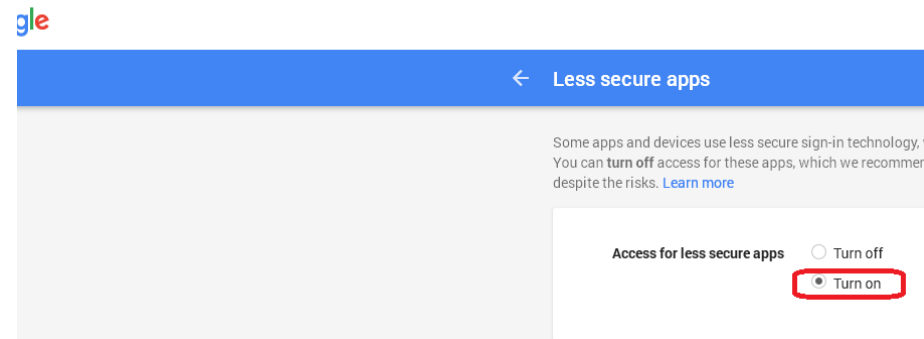
Mark Homans
357　1　3　12

For the other answers to work "from a server" first **Turn On Access
for less secure apps** in the gmail account.

83

Looks like recently google changed it's security policy. The top rated answer no longer works, until you change your account settings as described here:

https://support.google.com/accounts/answer/6010255?hl=en-GB

https://www.google.com/settings/security/lesssecureapps

gle

As of March 2016, google changed the setting location again!

edited Mar 22 '18 at 13:03

rogerdpack
35.4k 17 137 258

answered Sep 8 '15 at 12:09

BCS Software
919 7 7

2   This worked for me. And is also concerning. Not sure I want to turn that security off. May need to rethink... – Sully Apr 13 '16 at 12:41

2   From security point of view better to turn on 2-step Verification and then generate and use app password- see How to send an email in .Net according to new security policies? – Michael Freidgeim Jun 5 '16 at 12:57

1   @BCS Software, inmy program, the user insert any email which my program has to use it to send the message throught. So, how I can make the email user able to send the email even if the 2-factor authentication is turned on?? – Alaa' Jan 18 '18 at 13:44

▲

18

▼

Google may block sign in attempts from some apps or devices that do not use modern security standards. Since these apps and devices are easier to break into, blocking them helps keep your account safer.

Some examples of apps that do not support the latest security standards include:

- The Mail app on your iPhone or iPad with iOS 6 or below
- The Mail app on your Windows phone preceding the 8.1 release
- Some Desktop mail clients like Microsoft Outlook and Mozilla Thunderbird

Therefore, you have to enable **Less Secure Sign-In** in your google account.

After sign into google account, go to:

https://myaccount.google.com/lesssecureapps
or
https://www.google.com/settings/security/lesssecureapps

In C#, you can use the following code:

```csharp
using (MailMessage mail = new MailMessage())
{
    mail.From = new MailAddress("email@gmail.com");
    mail.To.Add("somebody@domain.com");
    mail.Subject = "Hello World";
    mail.Body = "<h1>Hello</h1>";
    mail.IsBodyHtml = true;
    mail.Attachments.Add(new Attachment("C:\\file.zip"));

    using (SmtpClient smtp = new SmtpClient("smtp.gmail.com", 587))
    {
        smtp.Credentials = new NetworkCredential("email@gmail.com",
        smtp.EnableSsl = true;
        smtp.Send(mail);
    }
}
```

edited Sep 13 '17 at 2:40

answered Aug 9 '14 at 7:28

mjb
3,228   5   28   43

Try This,

4

```csharp
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        MailMessage mail = new MailMessage();
        SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com")

        mail.From = new MailAddress("your_email_address@gmail.com
        mail.To.Add("to_address");
        mail.Subject = "Test Mail";
        mail.Body = "This is for testing SMTP mail from GMAIL";
```

```csharp
                    SmtpServer.Port = 587;
                    SmtpServer.Credentials = new System.Net.NetworkCredential
         "password");
                    SmtpServer.EnableSsl = true;

                    SmtpServer.Send(mail);
                    MessageBox.Show("mail Send");
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.ToString());
                }
            }
        }
```

answered Jan 31 '17 at 8:23

Here is my version: "[Send Email In C # Using Gmail](#)".

15

```csharp
using System;
using System.Net;
using System.Net.Mail;

namespace SendMailViaGmail
{
    class Program
    {
    static void Main(string[] args)
    {

        //Specify senders gmail address
        string SendersAddress = "Sendersaddress@gmail.com";
        //Specify The Address You want to sent Email To(can be any val
        string ReceiversAddress = "ReceiversAddress@yahoo.com";
        //Specify The password of gmial account u are using to sent ma
sender@gmail.com)
        const string SendersPassword = "Password";
        //Write the subject of ur mail
```

```csharp
            const string subject = "Testing";
            //Write the contents of your mail
            const string body = "Hi This Is my Mail From Gmail";

            try
            {
                //we will use Smtp client which allows us to send email using
                //i have specified the properties of SmtpClient smtp within{
                //gmails smtp server name is smtp.gmail.com and port number
                SmtpClient smtp = new SmtpClient
                {
                    Host = "smtp.gmail.com",
                    Port = 587,
                    EnableSsl = true,
                    DeliveryMethod = SmtpDeliveryMethod.Network,
                    Credentials    = new NetworkCredential(SendersAddress, Se
                    Timeout = 3000
                };

                //MailMessage represents a mail message
                //it is 4 parameters(From,TO,subject,body)

                MailMessage message = new MailMessage(SendersAddress, Receiv
body);
                /*WE use smtp sever we specified above to send the message(M
message)*/

                smtp.Send(message);
                Console.WriteLine("Message Sent Successfully");
                Console.ReadKey();
            }

            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
                Console.ReadKey();
            }
        }
    }
}
```

edited Jul 28 '16 at 8:43

Sujania
3,093   6   31   59

answered Oct 17 '10 at 21:19

tehie
159   1   2

7   While your article may in fact answer the question, <u>it would be preferable</u> to include the essential parts of the answer here, and provide the link for reference. Stack Overflow is only as useful as its questions and answers, and if your blog host goes down or your URLs get moved around, this answer becomes useless. Thanks! – sarnold Jan 16 '12 at 6:08 ✏

Try this one

1

```csharp
public static bool Send(string receiverEmail, string ReceiverName, s
string body)
{
        MailMessage mailMessage = new MailMessage();
        MailAddress mailAddress = new MailAddress("abc@gmail.com", ":
abc@gmail.com = input Sender Email Address
        mailMessage.From = mailAddress;
        mailAddress = new MailAddress(receiverEmail, ReceiverName);
        mailMessage.To.Add(mailAddress);
        mailMessage.Subject = subject;
        mailMessage.Body = body;
        mailMessage.IsBodyHtml = true;

        SmtpClient mailSender = new SmtpClient("smtp.gmail.com", 587
        {
            EnableSsl = true,
            UseDefaultCredentials = false,
            DeliveryMethod = System.Net.Mail.SmtpDeliveryMethod.Netw
            Credentials = new NetworkCredential("abc@gmail.com", "pa:
abc@gmail.com = input sender email address

sender email password
        };

        try
        {
            mailSender.Send(mailMessage);
            return true;
```

```csharp
        }
        catch (SmtpFailedRecipientException ex)
        { }
        catch (SmtpException ex)
        { }
        finally
        {
            mailSender = null;
            mailMessage.Dispose();
        }
        return false;
    }
```

answered Feb 27 '16 at 12:40

reza.cse08
3,679   28   27

```csharp
using System;
using System.Net;
using System.Net.Mail;

namespace SendMailViaGmail
{
    class Program
    {
    static void Main(string[] args)
    {

        //Specify senders gmail address
        string SendersAddress = "Sendersaddress@gmail.com";
        //Specify The Address You want to sent Email To(can be any val
        string ReceiversAddress = "ReceiversAddress@yahoo.com";
        //Specify The password of gmial account u are using to sent ma
sender@gmail.com)
        const string SendersPassword = "Password";
        //Write the subject of ur mail
        const string subject = "Testing";
        //Write the contents of your mail
        const string body = "Hi This Is my Mail From Gmail";
```

```csharp
        try
        {
            //we will use Smtp client which allows us to send email using
            //i have specified the properties of SmtpClient smtp within{
            //gmails smtp server name is smtp.gmail.com and port number
            SmtpClient smtp = new SmtpClient
            {
                Host = "smtp.gmail.com",
                Port = 587,
                EnableSsl = true,
                DeliveryMethod = SmtpDeliveryMethod.Network,
                Credentials = new NetworkCredential(SendersAddress, Sende
                Timeout = 3000
            };

            //MailMessage represents a mail message
            //it is 4 parameters(From,TO,subject,body)

            MailMessage message = new MailMessage(SendersAddress, Receiv
body);
            /*WE use smtp sever we specified above to send the message(M
message)*/

            smtp.Send(message);
            Console.WriteLine("Message Sent Successfully");
            Console.ReadKey();
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
            Console.ReadKey();
        }
    }
}
}
```

answered Oct 10 '15 at 13:57

Moin Shirazi

3,752   2   19   35

One Tip! Check the sender inbox, maybe you need allow less secure

4

apps. See: https://www.google.com/settings/security/lesssecureapps

answered Jul 21 '15 at 18:53

Gustavo Rossi Muller
486   5   11

---

use this way

4

```
MailMessage sendmsg = new MailMessage(SendersAddress, ReceiversAddres
SmtpClient client = new SmtpClient("smtp.gmail.com");

client.Port = Convert.ToInt32("587");
client.EnableSsl = true;
client.Credentials = new System.Net.NetworkCredential("mail-id@gmail
client.Send(sendmsg);
```

Don't forget this :

```
using System.Net;
using System.Net.Mail;
```

answered Jul 9 '15 at 6:57

alireza amini
1,111   1   10   28

I had the same issue, but it was resolved by going to gmail's security settings and **Allowing Less Secure apps**. The Code from Domenic & Donny works, but only if you enabled that setting

3

If you are signed in (to Google) you can follow this link and toggle **"Turn on"** for **"Access for less secure apps"**

answered Jun 25 '15 at 7:09

The problem for me was that my **password had a blackslash "\"** in it, which I copy pasted without realizing it would cause problems.

1

answered Jun 15 '15 at 11:34

Satbir Kira
409 3 17

Include this,

8

```
using System.Net.Mail;
```

And then,

```
MailMessage sendmsg = new MailMessage(SendersAddress, ReceiversAddre
SmtpClient client = new SmtpClient("smtp.gmail.com");

client.Port = Convert.ToInt16("587");
client.Credentials = new System.Net.NetworkCredential("mail-id@gmail
client.EnableSsl = true;

client.Send(sendmsg);
```

edited Aug 16 '14 at 9:59

Sam
5,535 8 36 61

answered Oct 9 '13 at 12:39

GOPI
832 7 26

I hope this code will work fine. You can have a try.

13

```csharp
// Include this.
using System.Net.Mail;

string fromAddress = "xyz@gmail.com";
string mailPassword = "*****";        // Mail id password from where i
string messageBody = "Write the body of the message here.";


// Create smtp connection.
SmtpClient client = new SmtpClient();
client.Port = 587;//outgoing port for the mail.
client.Host = "smtp.gmail.com";
client.EnableSsl = true;
client.Timeout = 10000;
client.DeliveryMethod = SmtpDeliveryMethod.Network;
client.UseDefaultCredentials = false;
client.Credentials = new System.Net.NetworkCredential(fromAddress, ma


// Fill the mail form.
var send_mail = new MailMessage();

send_mail.IsBodyHtml = true;
//address from where mail will be sent.
send_mail.From = new MailAddress("from@gmail.com");
//address to which mail will be sent.
send_mail.To.Add(new MailAddress("to@example.com"));
//subject of the mail.
send_mail.Subject = "put any subject here";

send_mail.Body = messageBody;
client.Send(send_mail);
```

edited Aug 16 '14 at 9:56

Sam
5,535 8 36 61

answered Oct 15 '13 at 10:00

Premdeep Mohanty
731 5 10

1   message send_mail = new MailMessage(); How is this line suppose to
     work? You can't implicitly convert 'System.Net.Mail.MailMessage' to
     'System.Windows.Forms.Message' – Debaprasad Mar 18 '14 at 6:23 ✏️

▲

2

▼

Here is one method to send mail and getting credentials from
web.config:

```
public static string SendEmail(string To, string Subject, string Msg
false, bool test = false, Stream AttachmentStream = null, string Att
string AttachmentFileName = null)
{
    try
    {
        System.Net.Mail.MailMessage newMsg = new
System.Net.Mail.MailMessage(System.Configuration.ConfigurationManager
 To, Subject, Msg);
        newMsg.BodyEncoding = System.Text.Encoding.UTF8;
        newMsg.HeadersEncoding = System.Text.Encoding.UTF8;
        newMsg.SubjectEncoding = System.Text.Encoding.UTF8;

        System.Net.Mail.SmtpClient smtpClient = new System.Net.Mail.
        if (AttachmentStream != null && AttachmentType != null && At
null)
        {
            System.Net.Mail.Attachment attachment = new
System.Net.Mail.Attachment(AttachmentStream, AttachmentFileName);
            System.Net.Mime.ContentDisposition disposition =
attachment.ContentDisposition;
            disposition.FileName = AttachmentFileName;
            disposition.DispositionType =
System.Net.Mime.DispositionTypeNames.Attachment;

            newMsg.Attachments.Add(attachment);
        }
        if (test)
        {
            smtpClient.PickupDirectoryLocation = "C:\\TestEmail";
            smtpClient.DeliveryMethod =
System.Net.Mail.SmtpDeliveryMethod.SpecifiedPickupDirectory;
```

```
            }
            else
            {
                //smtpClient.EnableSsl = true;
            }

            newMsg.IsBodyHtml = bodyHtml;
            smtpClient.Send(newMsg);
            return SENT_OK;
        }
        catch (Exception ex)
        {

            return "Error: " + ex.Message
                + "<br/><br/>Inner Exception: "
                + ex.InnerException;
        }

    }
```

And the corresponding section in web.config:

```
<appSettings>
    <add key="mailCfg" value="yourmail@example.com"/>
</appSettings>
<system.net>
  <mailSettings>
    <smtp deliveryMethod="Network" from="yourmail@example.com">
      <network defaultCredentials="false" host="mail.exapmple.com"
userName="yourmail@example.com" password="your_password" port="25"/>
    </smtp>
  </mailSettings>
</system.net>
```

answered Oct 8 '13 at 6:10

iTURTEV
181   4

If you want to send background email, then please do the below

6

```
public void SendEmail(string address, string subject, string message
{
Thread threadSendMails;
threadSendMails = new Thread(delegate()
    {

        //Place your Code here

    });
  threadSendMails.IsBackground = true;
  threadSendMails.Start();
}
```

and add namespace

```
using System.Threading;
```

edited Oct 3 '13 at 18:03

Kevin Panko
6,999   9   44   55

answered Jul 23 '13 at 9:47

RAJESH KUMAR
307   2   12

This is to send email with attachement.. Simple and short..

39

source: http://coding-issues.blogspot.in/2012/11/sending-email-with-attachments-from-c.html

```
using System.Net;
using System.Net.Mail;

public void email_send()
```

```csharp
    {
        MailMessage mail = new MailMessage();
        SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com");
        mail.From = new MailAddress("your mail@gmail.com");
        mail.To.Add("to_mail@gmail.com");
        mail.Subject = "Test Mail - 1";
        mail.Body = "mail with attachment";

        System.Net.Mail.Attachment attachment;
        attachment = new System.Net.Mail.Attachment("c:/textfile.txt");
        mail.Attachments.Add(attachment);

        SmtpServer.Port = 587;
        SmtpServer.Credentials = new System.Net.NetworkCredential("your
"your password");
        SmtpServer.EnableSsl = true;

        SmtpServer.Send(mail);

    }
```

edited Jul 29 '13 at 13:10

answered May 28 '12 at 12:41

Ranadheer Reddy
2,625   10   42   69

Changing sender on Gmail / Outlook.com email:

3

To prevent spoofing - Gmail/Outlook.com won't let you send from an arbitrary user account name.

If you have a limited number of senders you can follow these instructions and then set the `From` field to this address: Sending mail from a different address

If you are wanting to send from an arbitrary email address (such as a feedback form on website where the user enters their email and you don't want them emailing you directly) about the best you can do is this :

```
msg.ReplyToList.Add(new System.Net.Mail.MailAddress(email, fr
```

This would let you just hit 'reply' in your email account to reply to the fan of your band on a feedback page, but they wouldn't get your actual email which would likely lead to a tonne of spam.

If you're in a controlled environment this works great, but please note that I've seen some email clients send to the from address even when reply-to is specified (I don't know which).

answered Jul 7 '13 at 20:49

Simon_Weaver
74.6k   63   460   531

The above answer doesn't work. You have to set `DeliveryMethod = SmtpDeliveryMethod.Network` or it will come back with a "**client was not authenticated**" error. Also it's always a good idea to put a timeout.

148

Revised code:

```
using System.Net.Mail;
using System.Net;

var fromAddress = new MailAddress("from@gmail.com", "From Name");
var toAddress = new MailAddress("to@yahoo.com", "To Name");
const string fromPassword = "password";
const string subject = "test";
const string body = "Hey now!!";

var smtp = new SmtpClient
```

```
{
    Host = "smtp.gmail.com",
    Port = 587,
    EnableSsl = true,
    DeliveryMethod = SmtpDeliveryMethod.Network,
    Credentials = new NetworkCredential(fromAddress.Address, fromPas
    Timeout = 20000
};
using (var message = new MailMessage(fromAddress, toAddress)
{
    Subject = subject,
    Body = body
})
{
    smtp.Send(message);
}
```

edited May 8 '13 at 4:16

nawfal
43.9k   36   258   305

answered Jan 28 '09 at 22:01

Donny V.
9,756   9   52   69

---

2   Interesting; it works on my machine (TM). But since this seems plausible, I'll add it to my answer. – Domenic Mar 16 '09 at 19:53

2   Hmm my guess is that SmtpDeliveryMethod.Network is the default, but maybe the default gets changed when running in IIS---was that what you were doing? – Domenic Mar 16 '09 at 19:55

4   I ran it from a desktop app. – Donny V. Mar 21 '09 at 2:47

2   I am using same code in Console application, it's through error **"Failure sending mail."** – Karthikeyan P Aug 26 '15 at 15:02

4   This answer doesnt work. Please, look at the question stackoverflow.com/questions/34851484/... – user5515846 Jan 19 '16 at 15:22

Source : [Send email in ASP.NET C#](#)

7

Below is a sample working code for sending in a mail using C#, in the below example I am using google's smtp server.

The code is pretty self explanatory, replace email and password with your email and password values.

```csharp
public void SendEmail(string address, string subject, string message
{
    string email = "yrshaikh.mail@gmail.com";
    string password = "put-your-GMAIL-password-here";

    var loginInfo = new NetworkCredential(email, password);
    var msg = new MailMessage();
    var smtpClient = new SmtpClient("smtp.gmail.com", 587);

    msg.From = new MailAddress(email);
    msg.To.Add(new MailAddress(address));
    msg.Subject = subject;
    msg.Body = message;
    msg.IsBodyHtml = true;

    smtpClient.EnableSsl = true;
    smtpClient.UseDefaultCredentials = false;
    smtpClient.Credentials = loginInfo;
    smtpClient.Send(msg);
}
```

answered Aug 22 '12 at 12:29

Yasser
29.8k   37   164   246

Instead of var ,I have used class name like NetworkCredential,MailMessage and SmtpClient.It work for me. – Jui Test
Feb 28 '13 at 7:33

protected by meagar ♦ Mar 22 '11 at 19:16

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?