

Misc

Send an Email from C#	2
Configure your gmail	2
Configure Visual Studio Project	6
Option 1: hardcode mail credentials	6
Send Email Code	6
Option 2: configure mail credentials	7
Send Email Code	7
Use SendEmail	9

Send an Email from C#

You may watch a short video explaining the details:

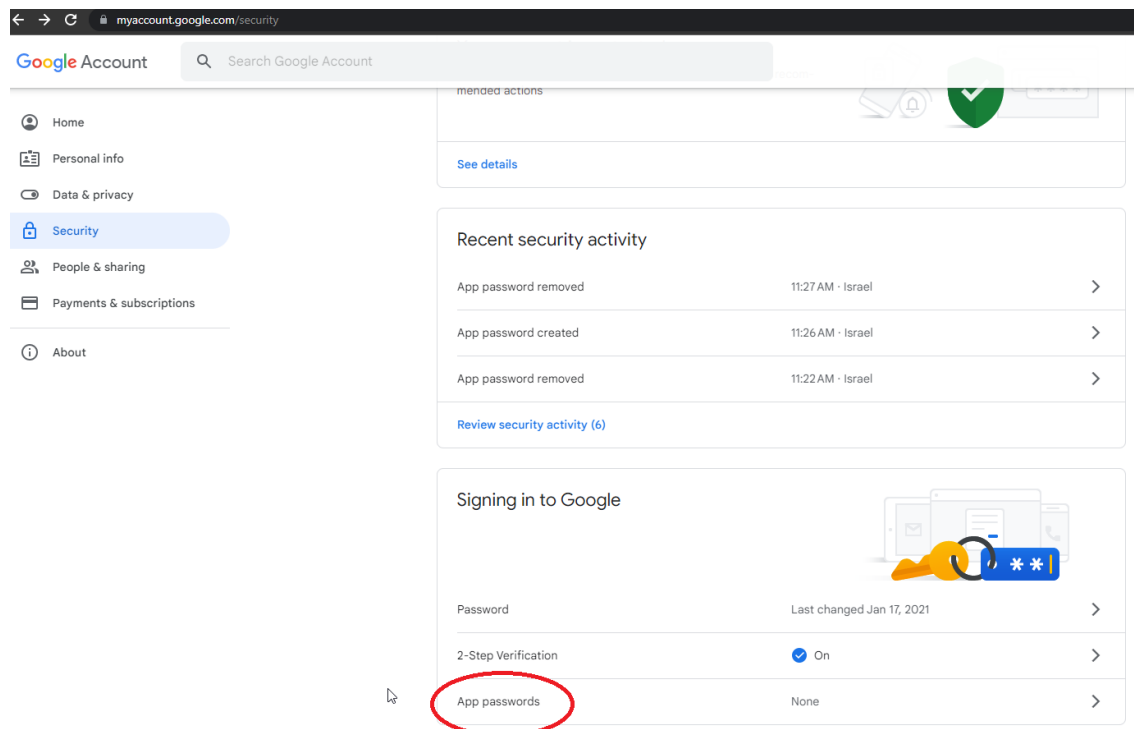
https://youtu.be/Ms_FDe2VzFU

We are going to utilize the gmail smtp server to send emails. However, first we need to do some configuration in our gmail account.

Configure your gmail

Visit <https://myaccount.google.com/security>

Under Signing in to Google, click: App Passwords



Select app → Select “Other (Custom name)”

← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

You don't have any app passwords.

Select the app and device you want to generate the app password for.

Select app



Select device



GENERATE

← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

You don't have any app passwords.

Select the app and device you want to generate the app password for.

Select app

Select device



Mail

Calendar

Contacts

YouTube

Other (Custom name)



GENERATE

Type your app name (Any free name)

← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

You don't have any app passwords.

Select the app and device you want to generate the app password for.

my awesome app ×

Click Generate

← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

You don't have any app passwords.

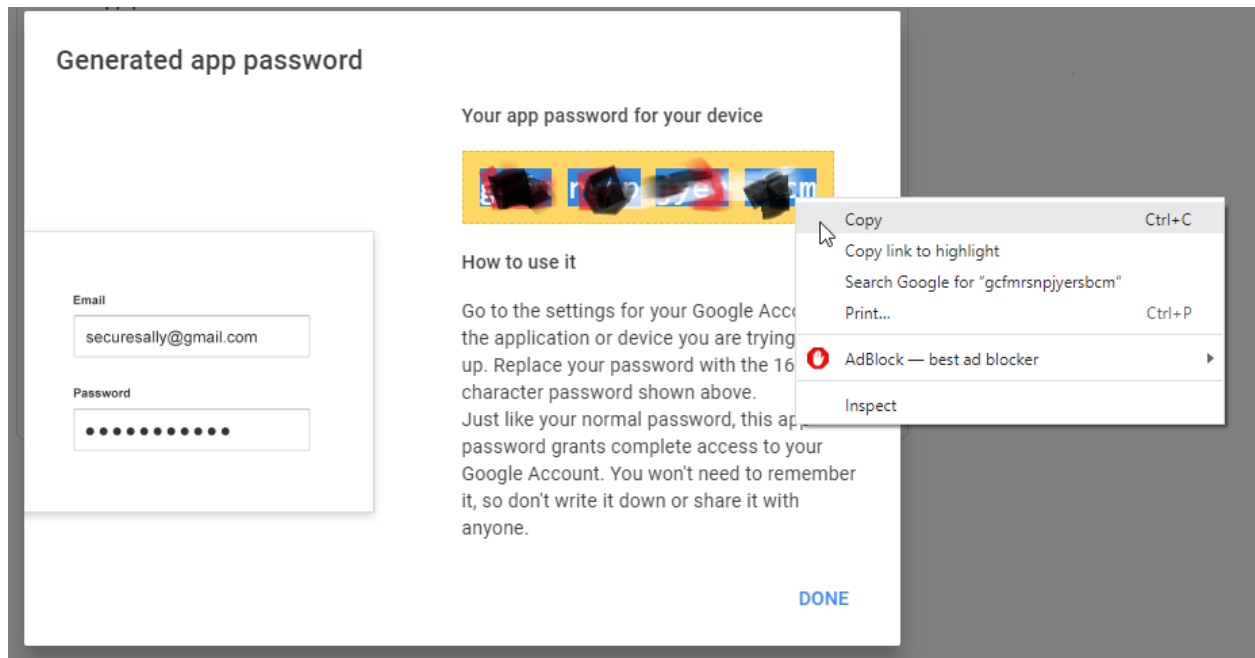
Select the app and device you want to generate the app password for.

my awesome app ×

GENERATE



Copy the password from the popup (Paste it somewhere - we need it later)



Click on "Done" button

Configure Visual Studio Project

Option 1: hardcode mail credentials

Send Email Code

Copy and Paste the following code to some class (could be new class - e.g., EmailUtils):

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

public static async Task SendEmail(string to, string subject, string body, bool
isBodyHTML = true)
{
    string from = "YOUR GMAIL";
    string display = "YOUR NAME";
    string emailPassword = "APP PASSWORD - from previous step";

    var fromEmail = new MailAddress(from, display);
    var toEmail = new MailAddress(to);
    var smtp = new SmtpClient
    {
        Host = "smtp.gmail.com",
        Port = 587,
        EnableSsl = true,
        Timeout = 60000, //60 seconds
        DeliveryMethod = SmtpDeliveryMethod.Network,
        UseDefaultCredentials = false,
        Credentials = new NetworkCredential(fromEmail.Address, emailPassword)
    };
    using var message = new MailMessage(fromEmail, toEmail)
    {
        Subject = subject,
        Body = body,
        IsBodyHtml = isBodyHTML,
    };
    try
    {
        await smtp.SendMailAsync(message);
    }
}
```

```

    catch
    {

    }
}

```

Option 2: configure mail credentials

We want to read the email credentials from some configuration file. In our case we are going to use **appsettings.json**, open the file and add the following:

```

"Email": {
  "from": {
    "mail": "YOUR GMAIL@gmail.com",
    "display": "YOUR NAME"
  },
  "password": "App password from previous section"
}

```

Note: modify the piece of code above to meet your details. Modify the following:

YOUR GMAIL

YOUR NAME

App password from previous section

Save file

Send Email Code

Same as previous one (hardcode mail) but this time we will take the credentials from the appsettings.json using IConfiguration:

```
using System.Net;
```

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

```
public static async Task SendEmail(IConfiguration config, string to, string subject, string
body, bool isBodyHTML = true)
```

```

{
    string? from = config.GetValue(typeof(string), "Email:from:mail")?.ToString();
    string? display = config.GetValue(typeof(string), "Email:from:display")?.ToString();
    string? emailPassword = config.GetValue(typeof(string),
"Email:password")?.ToString();
    if (emailPassword == null || from == null)

```

```
throw new Exception("Missing Email section in the appsettings");
```

```
var fromEmail = new MailAddress(from, display);
var toEmail = new MailAddress(to);
var smtp = new SmtpClient
{
    Host = "smtp.gmail.com",
    Port = 587,
    EnableSsl = true,
    Timeout = 5000,
    DeliveryMethod = SmtpDeliveryMethod.Network,
    UseDefaultCredentials = false,
    Credentials = new NetworkCredential(fromEmail.Address, emailPassword)
};
using var message = new MailMessage(fromEmail, toEmail)
{
    Subject = subject,
    Body = body,
    IsBodyHtml = isBodyHTML,
};
try
{
    await smtp.SendMailAsync(message);
}
catch
{
}
}
```


Use SendEmail

Locate the place you want to send email and type the following - this is an example, you may modify the code based on your use case. Assume we are willing to send the email once the user is Registered:

```
public async Task<ActionResult> Register(AppUser model)
{
    //save the user in the db
    ...
    //send him an email
    string subject = "Welcome to zzz Store.";
    string body = "<br/><br/><h3>Welcome, user.FirstName + ", " +
user.LastName!</h3><br/>" +
        "Thanks for your registration to our web site." +
        "<br/><br/>If you didn't register to our site, please send us an email"

    await EmailUtils.SendEmail(user.Email, subject, body, true);
    //In case of passing configuration
    //await EmailUtils.SendEmail(this.configuration, user.Email, subject, body, true);
}
```