Table of Contents

[Section 14: Deployment & Email 2](#_Toc145683541)

[14.1 Database Initializer 2](#_Toc145683542)

[14.2 Email Service 5](#_Toc145683543)

[14.2.1 Create SendGrid Account 5](#_Toc145683544)

[14.2.2 SendGrid Sender Authentication and API Key 10](#_Toc145683545)

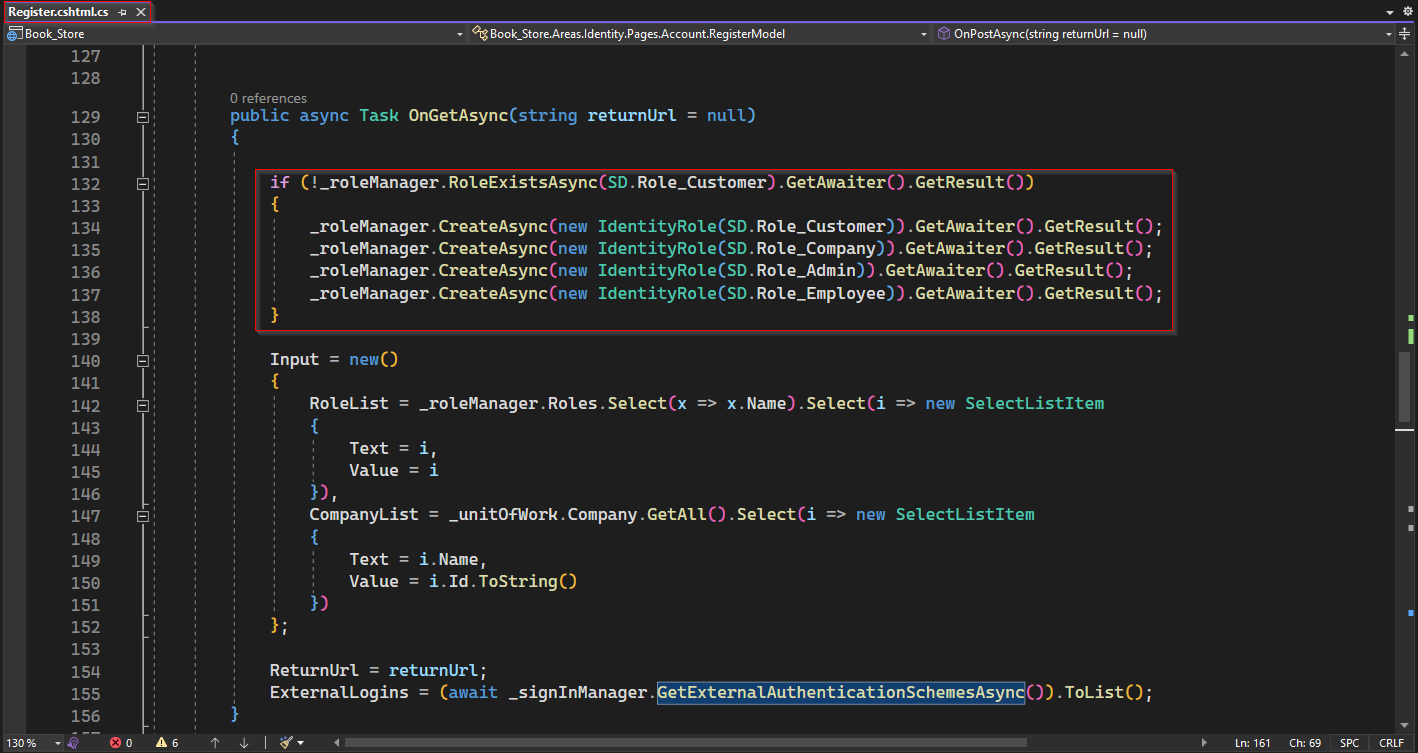
[14.2.3 Configure Email Service 16](#_Toc145683546)

[14.2.4 Use Email Service 18](#_Toc145683547)

# Section 14: Deployment & Email

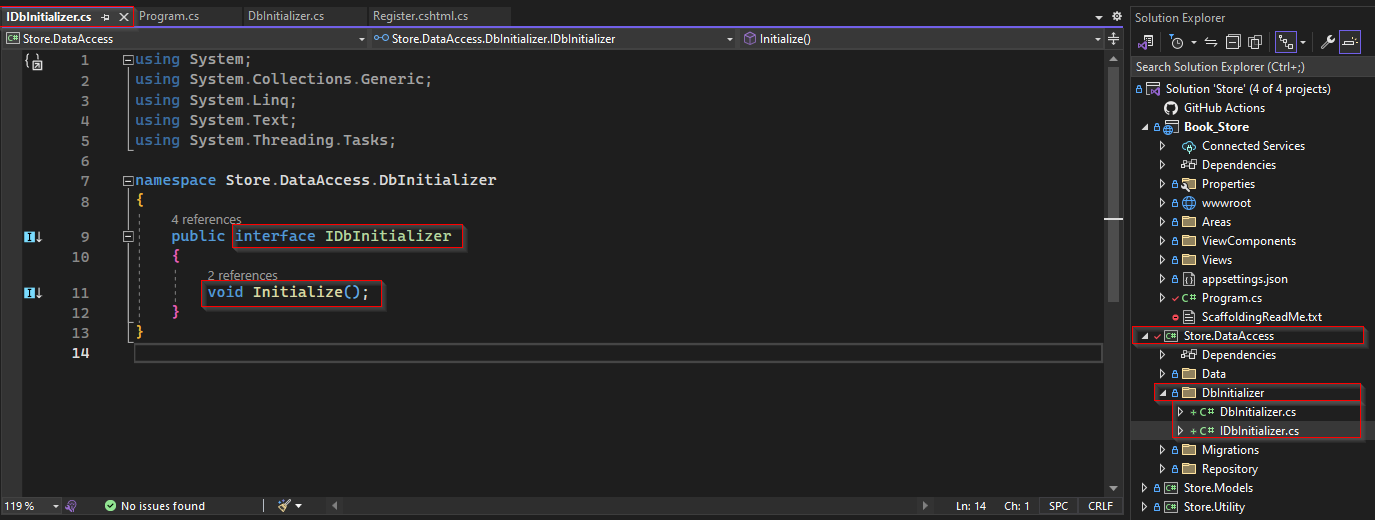
## 14.1 Database Initializer

Before we place this application in production, we need to think that how roles and the very first Admin user will be created. Previous we insert roles manually inside action method OnGetAsync of page model of Register page which is not a correct way.



Let’s explore correct and more professional way to seed the initial data to database.

First Create a folder "DbInitializer" inside DataAccess project. And inside the folder "DbInitializer" create an interface with name "IDbInitializer.cs" with a void method Initializer();

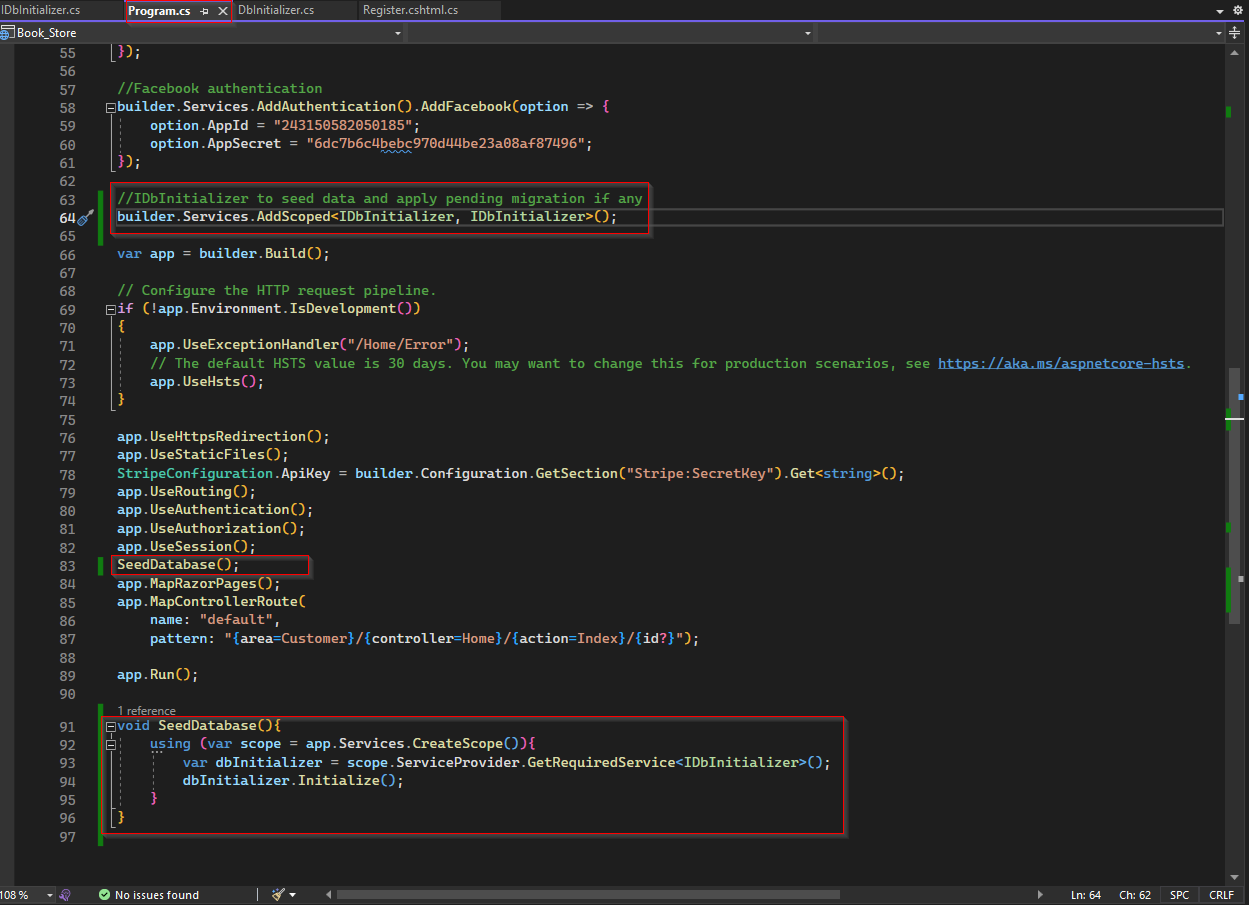


Now create a class "DbInitializer.cs" that inherit "IDbInitializer.cs". Inside this under implementation of Initializer(). We perform following three tasks

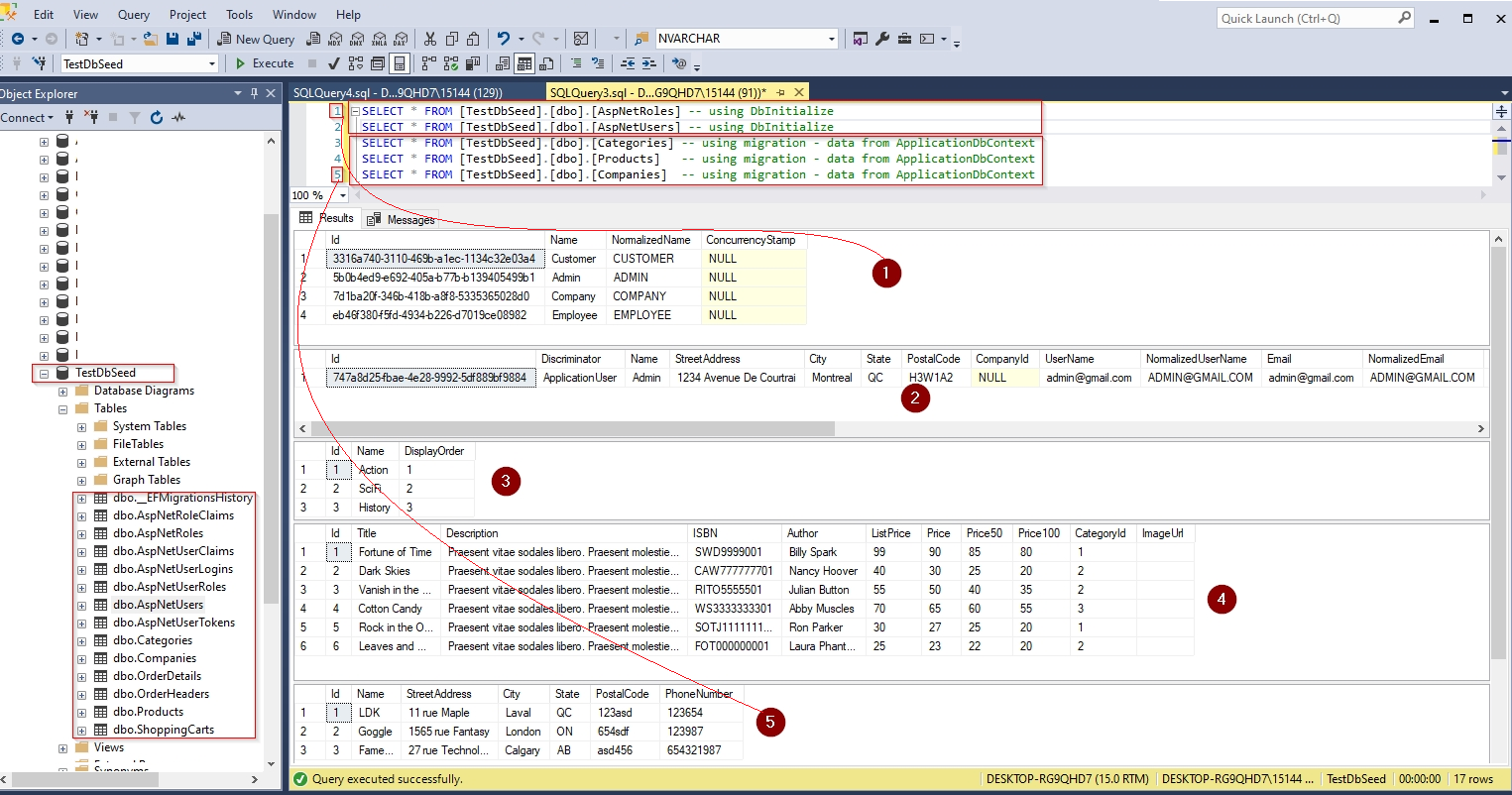
* Check for pending migration if any found push it to update database.
* Create roles if they are not created.
* Create the very first user with Admin role.



Then inside Program.cs configure Service for interface IDbInitilizer. And add it on pipeline so each time application run it will implement DbInitializer.



Let’s perform a test if everything working correctly. For this inside appsetting.json let’s modify the database name as TestDbSeed and run application. If everything worked as expected it should create database and seed roles data, Admin user, all migrations that are inside Migration folder.



## 14.2 Email Service

### 14.2.1 Create SendGrid Account

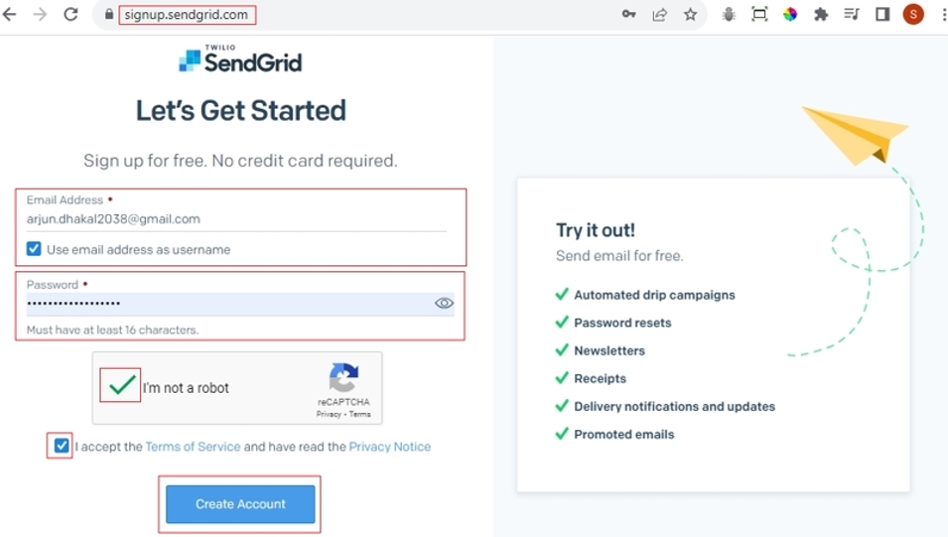
Using SendGrid we can send multiple emails without having an email server. It provides two ways to send emails.

* using SMTP relay
* using Web API

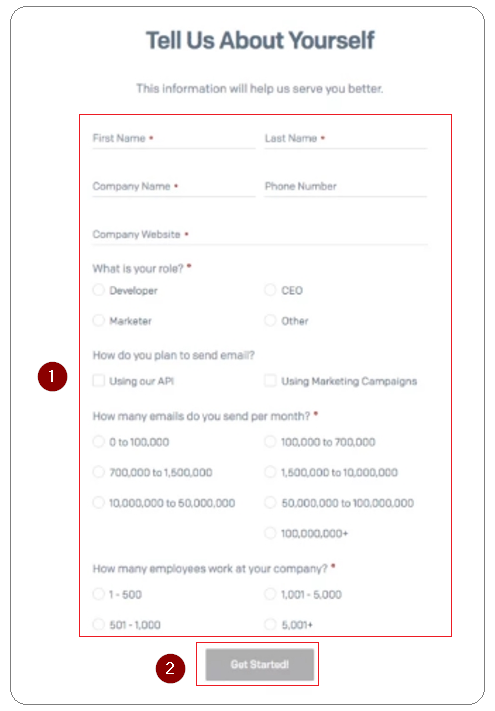
In this section we are going to use its Web API and for that we first need to have an account in SendGrid, where we create and set sender email, create an API Key, and using that API Key we can configure our email service in our application. But to use SendGrid for email, you must have a domain email, the personal email like Gmail, Yahoo etc. will not work.

To open account in SendGrid let go to its website typing sendgrid.com in browser.

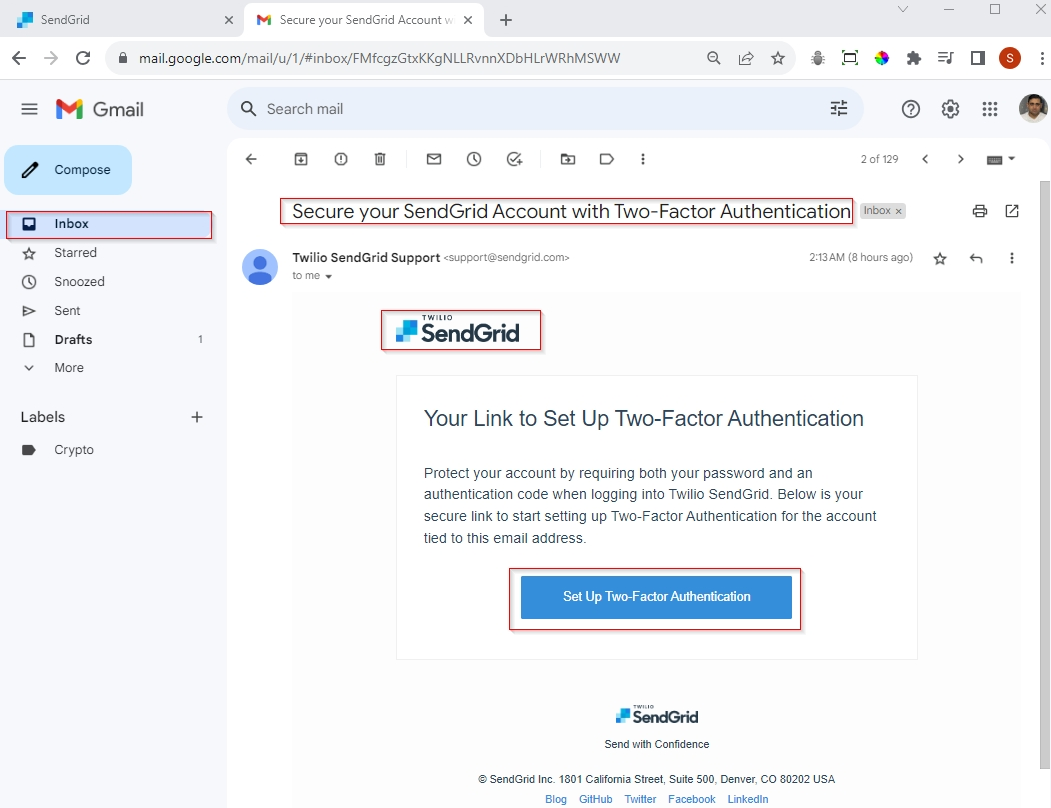


Then Click on "Start for free" button, will redirect to its signup page as below.

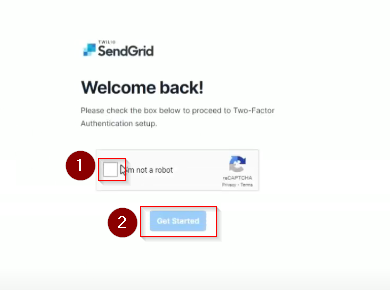
To sign up, provide an email address and password, then check the checkboxes for "I’m not a robot" and "I accept the Terms of Service and have read the Privacy Notice". Finally click "Create Account", it will take you to following page.



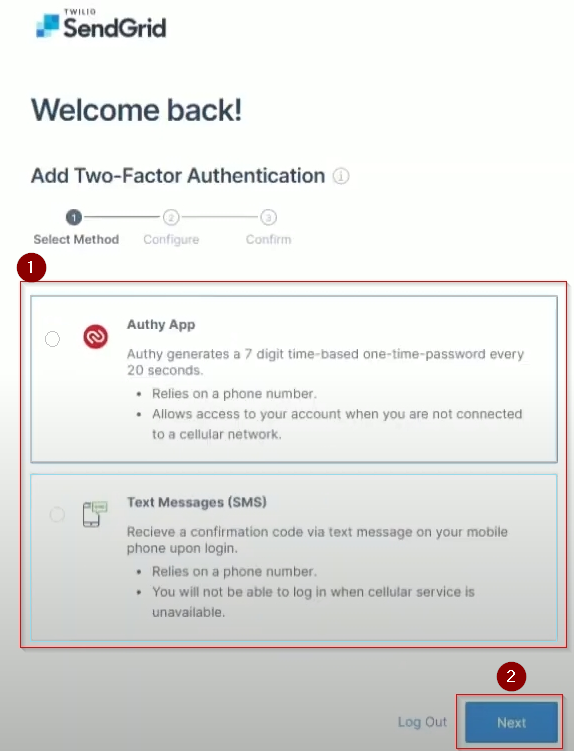
Once you provide values to all above fields, it will ask to set Two-Factor Authentication(2FA) sending a link to email address used for signup.



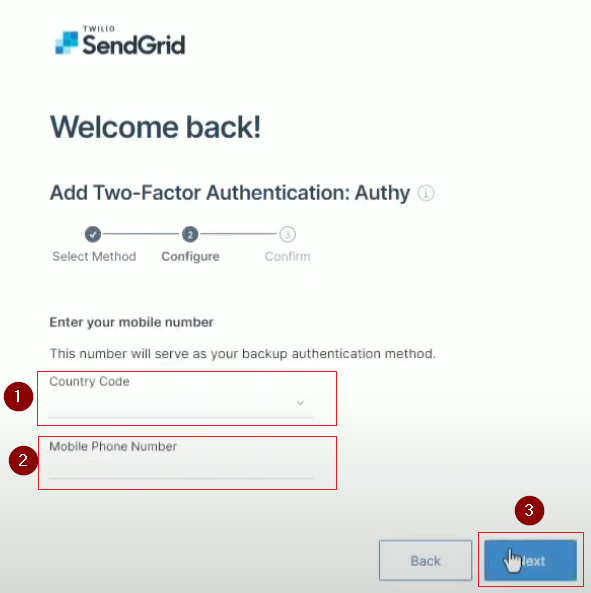
When you click "Set Up Two-Factor Authentication" button, it will redirect you to your SendGrid page as shown below.



Here check the checkbox "I’m not a robot" and click "Get Started", it will redirect to page to add two factor authentication setup page.



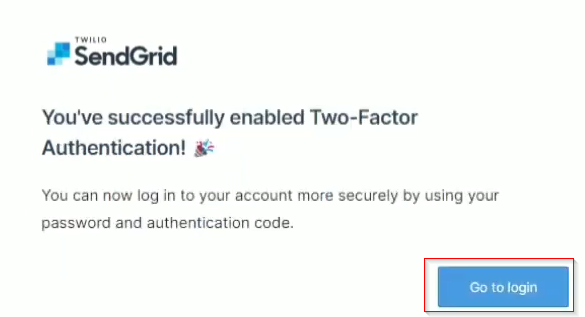
Here, you must choose an authentication option then click Next. If you choose first option, you will need to use a third-party App for OTP and if you choose SMS option you need to provide phone number in form below and hit "Next".



Once you received OTP on your phone, type it to form below then click "Save".



Hence account signup and two factor authentication (2FA) are completed. Click on "Go to login" it will redirect you to login page.

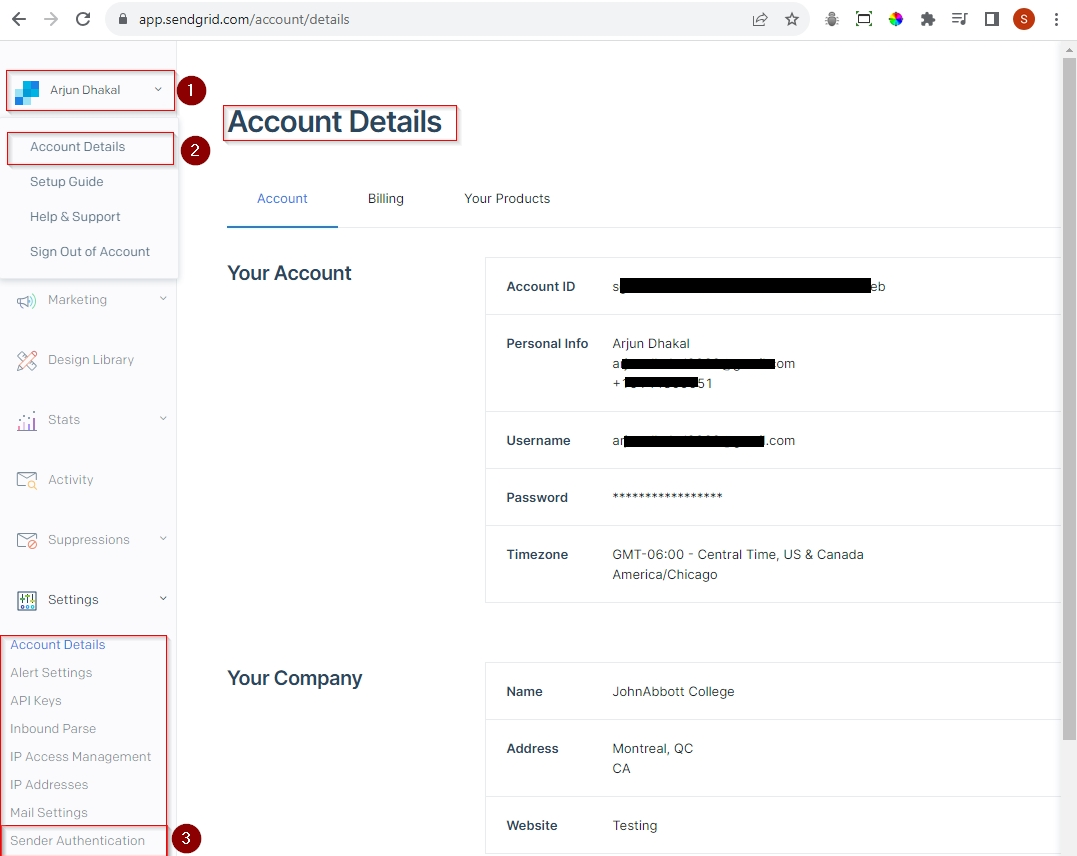


Once you provide your credentials it will send you OTP for 2FA. If you correctly provide OTP, it will take you your SendGrid Dashboard.

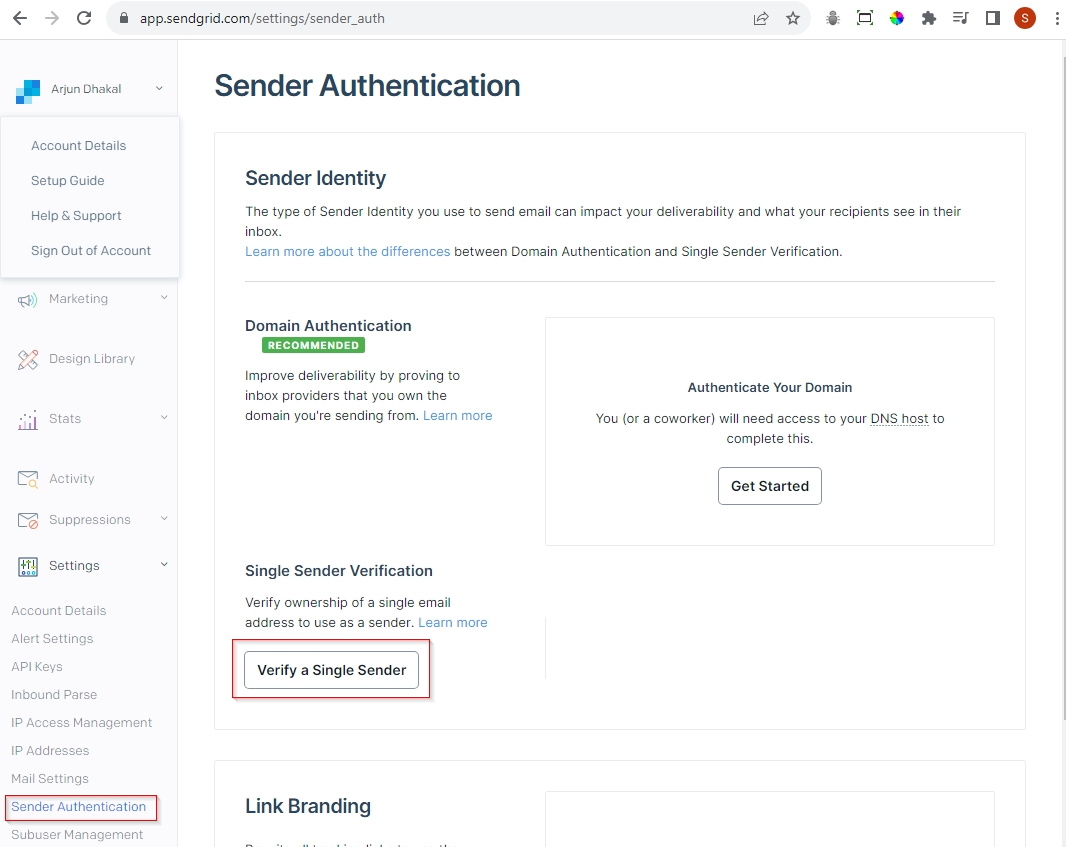
### 14.2.2 SendGrid Sender Authentication and API Key

Once you login to you SendGrid Account,

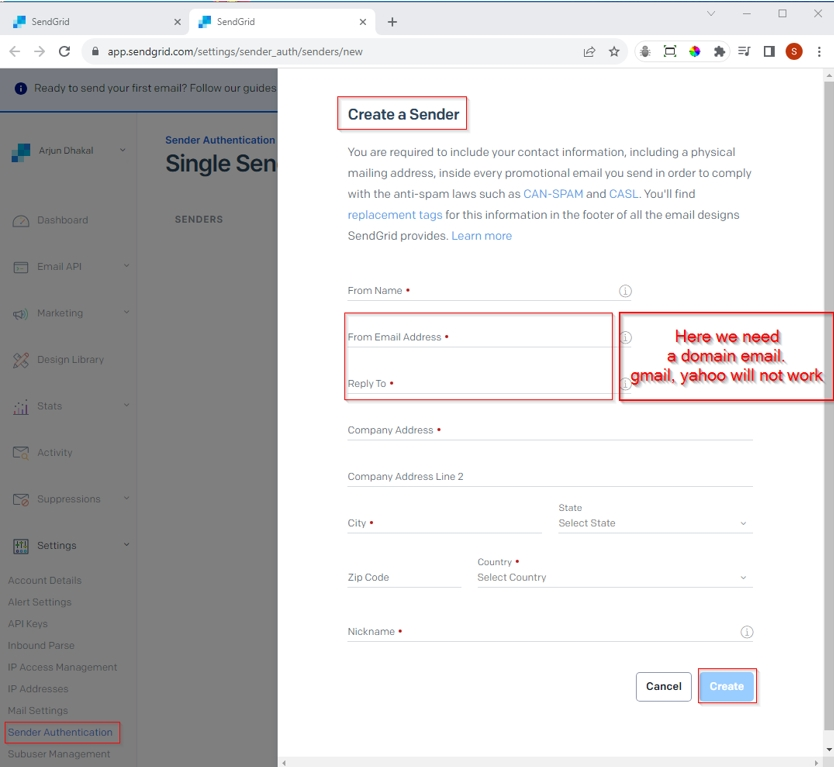
* click on dropdown list on your profile name.
* select Account Details
* Then click on Sender Authentication



Once you click on "Sender Authentication" it will take you to Sender Authentication page as shown below.

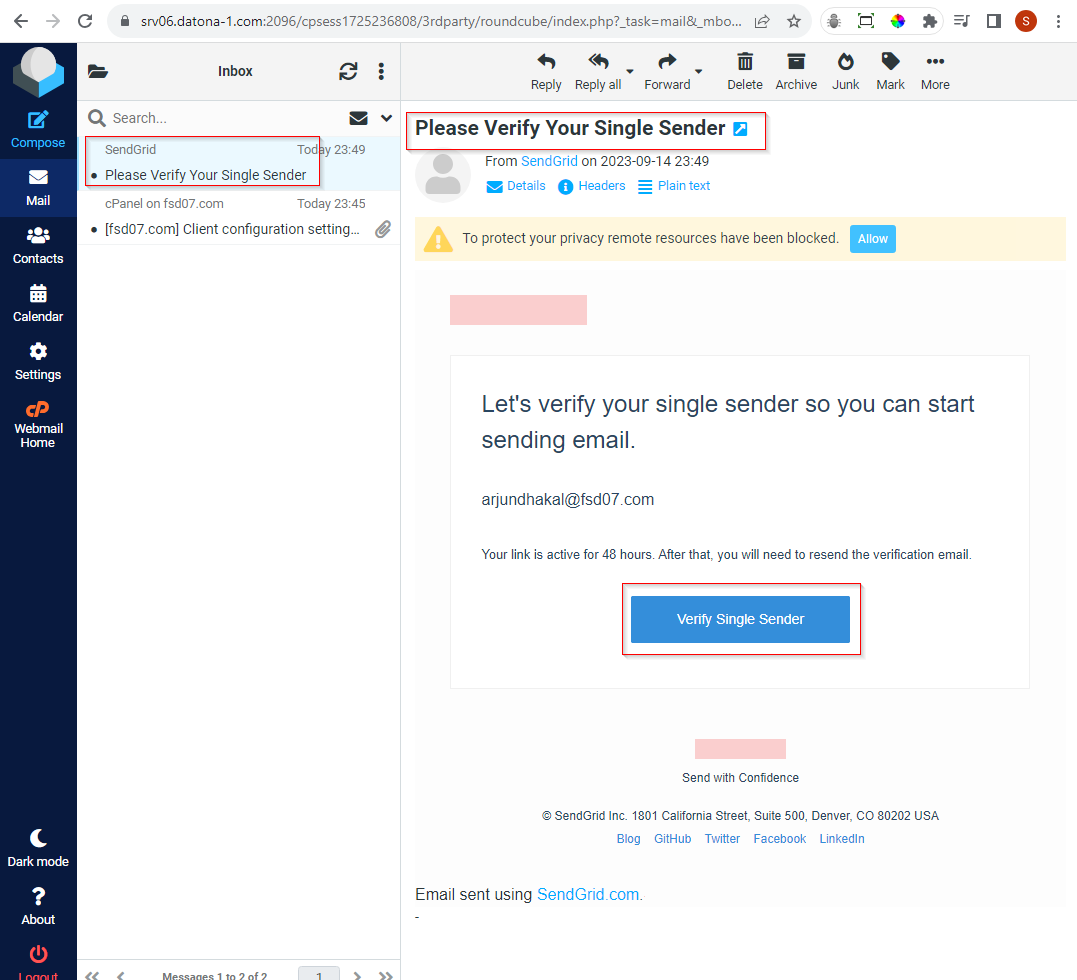


Here Click on "Verify a Single Sender", a Create Sender Form appears as shown below.

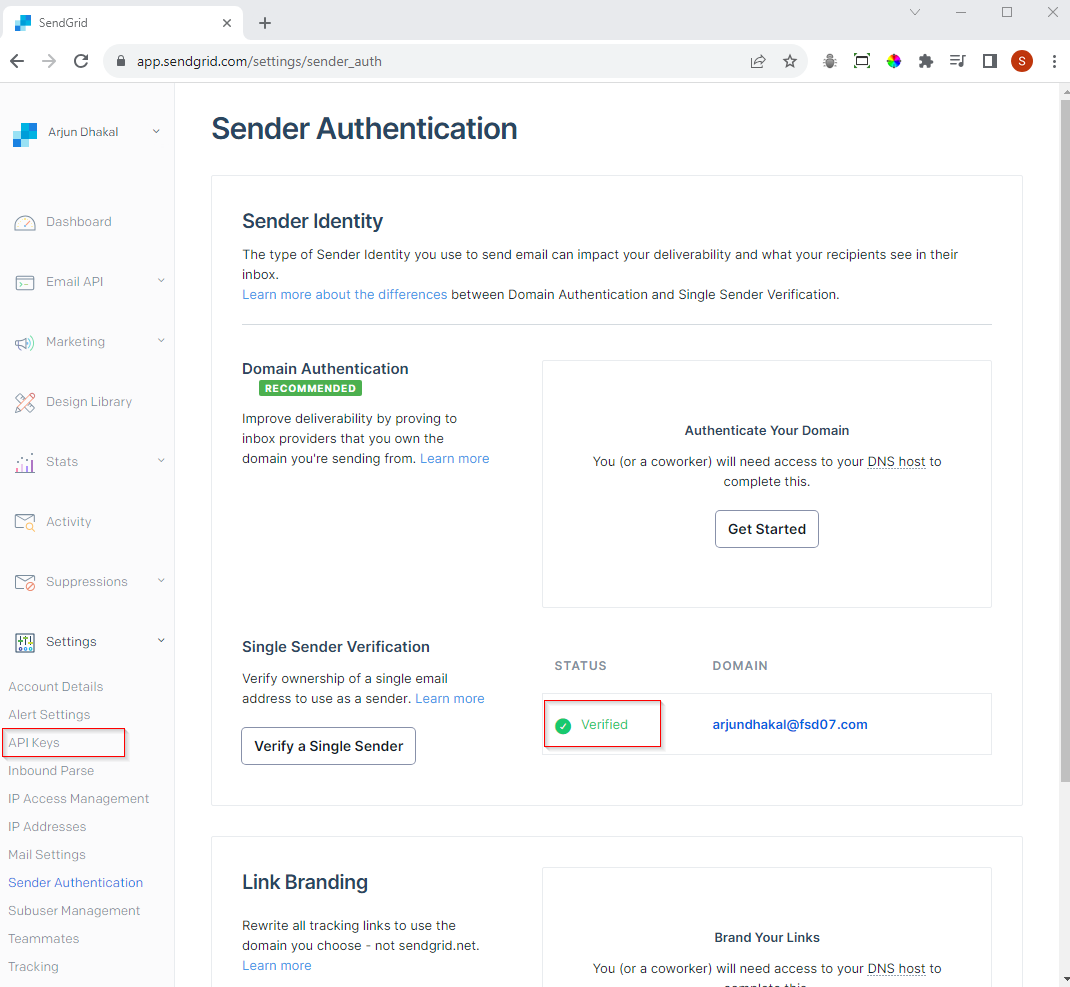


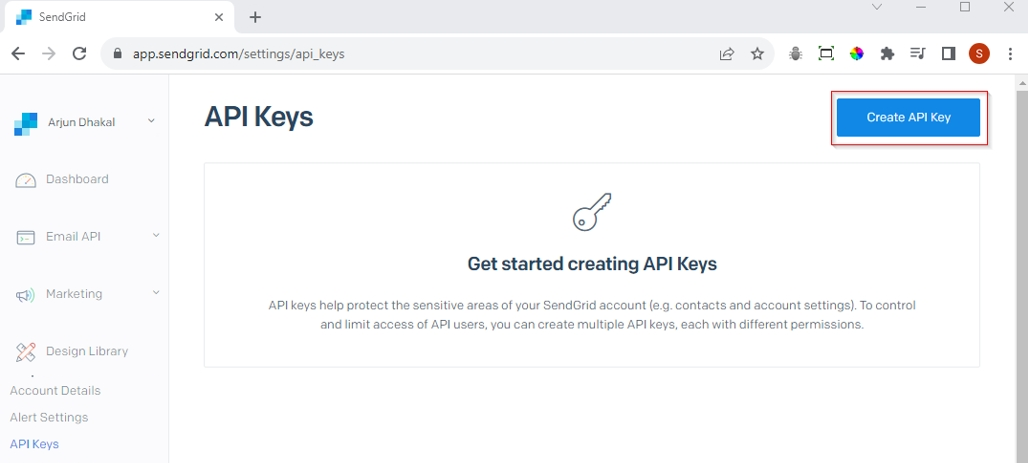
Once you fill up all required input field hit "Create" button. It will sent a verification to the email address you have provided here.

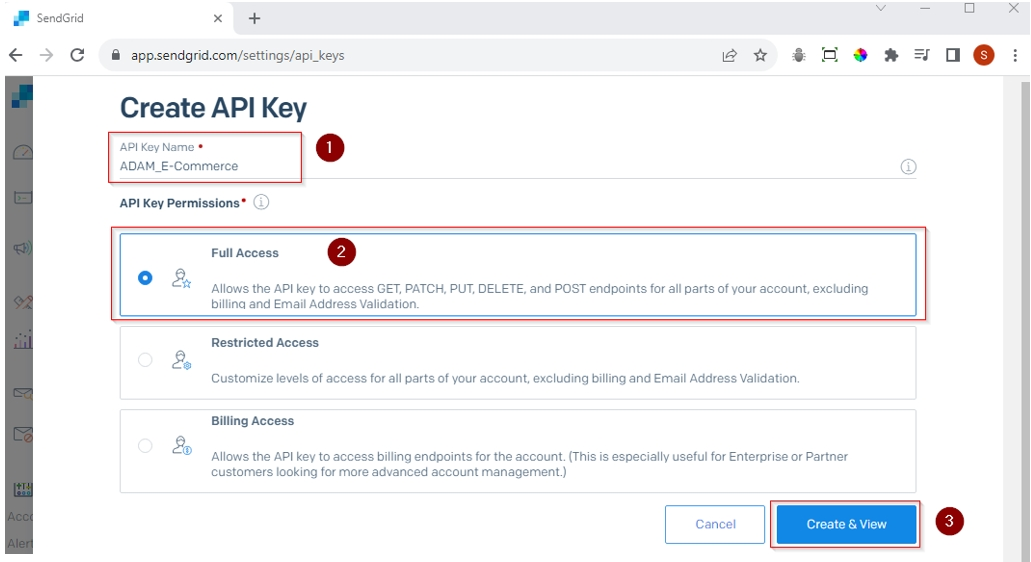
**Note: For the email address used to send email and receive both need to be a domain email. Personal email like Gmail or Yahoo doesn’t work.**



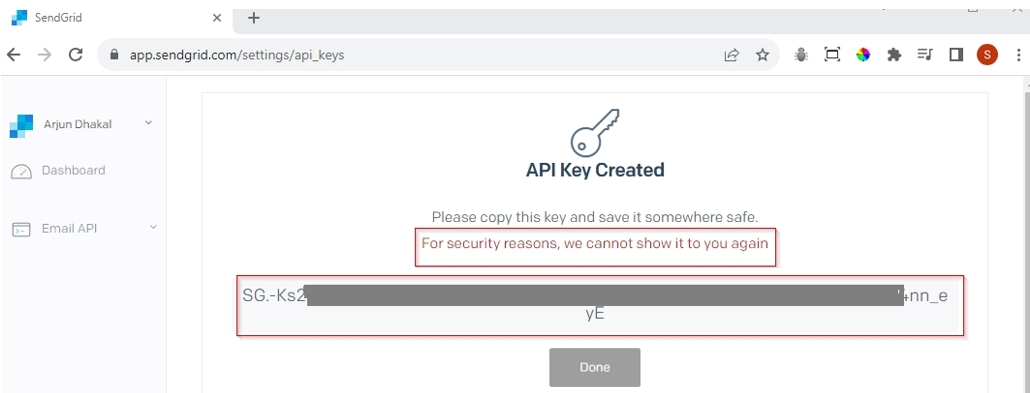
Here you need to click "Verify Single Sender" to verify. Once you successfully verified you can see the status of the sender authentication as below.



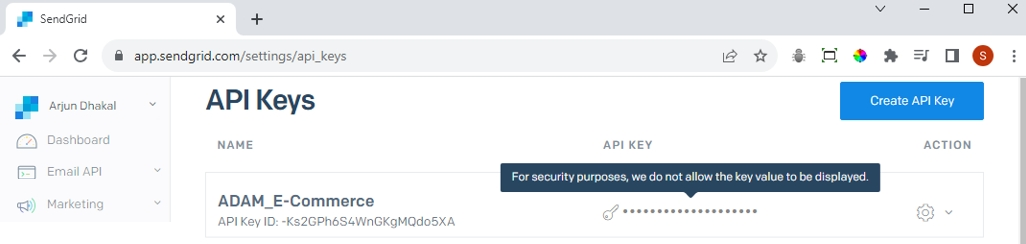
Here status of your domain is verified, and you can create an API Key, for that click on "API Keys", it will redirect to page as given below. Now click on "Create API Key" button



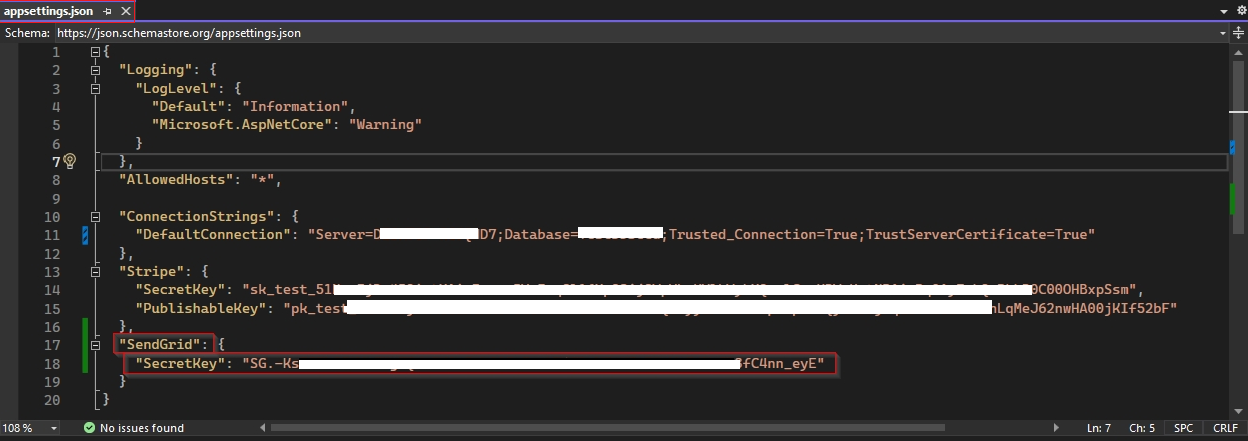
Here, you need to give a name to this API key, select Full Access and click Create & View.



Once API key is created, copy the key, and save in somewhere safe and secure place. Once you close this window you can view it again. We need this key to configure email service in our web application.



### 14.2.3 Configure Email Service

Now once we have the API key, we can add it on appsetting.json file of our application.

Next, we also need to install package for SendGrid on Utility project.

Now, we need to modify EmailSender.cs under Utility. To send email, public class EmailSender must have inherit from base class IEmailSender.

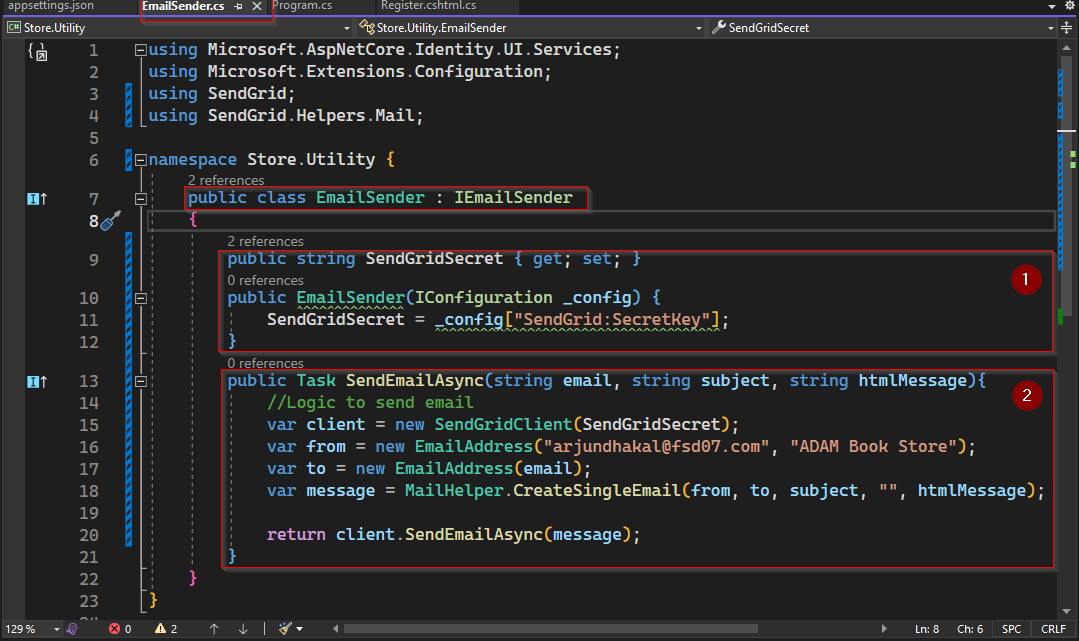
Now, let’s add a property SendGridSecret property of string type and to get SendGridSecret key from appsetting.json we need to inject IConfiguration interface.

The method SendEmailAsync has three parameters which will be supplied while calling this method, but beside those we use SendGrid helpers to grab following parameters.

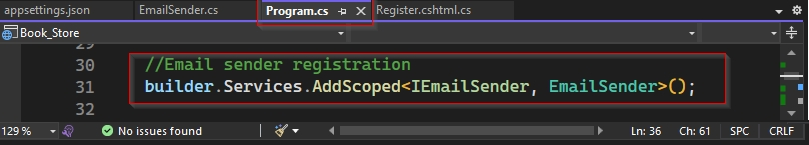
* Client using SendGrindClient passing SendGridSecret
* From email address using SendGrid EmailAddress function passing domain email address and email header
* To email address using SendGrid EmailAddress function passing input parameter email of type string

Then we can create message using SendGrid.MailHelper.CreateSingleEmail passing above parameter as shown in snippet below. Note for plain text we pass empty space.

Finally we return client.SendEmailAsync passing above message we just created.

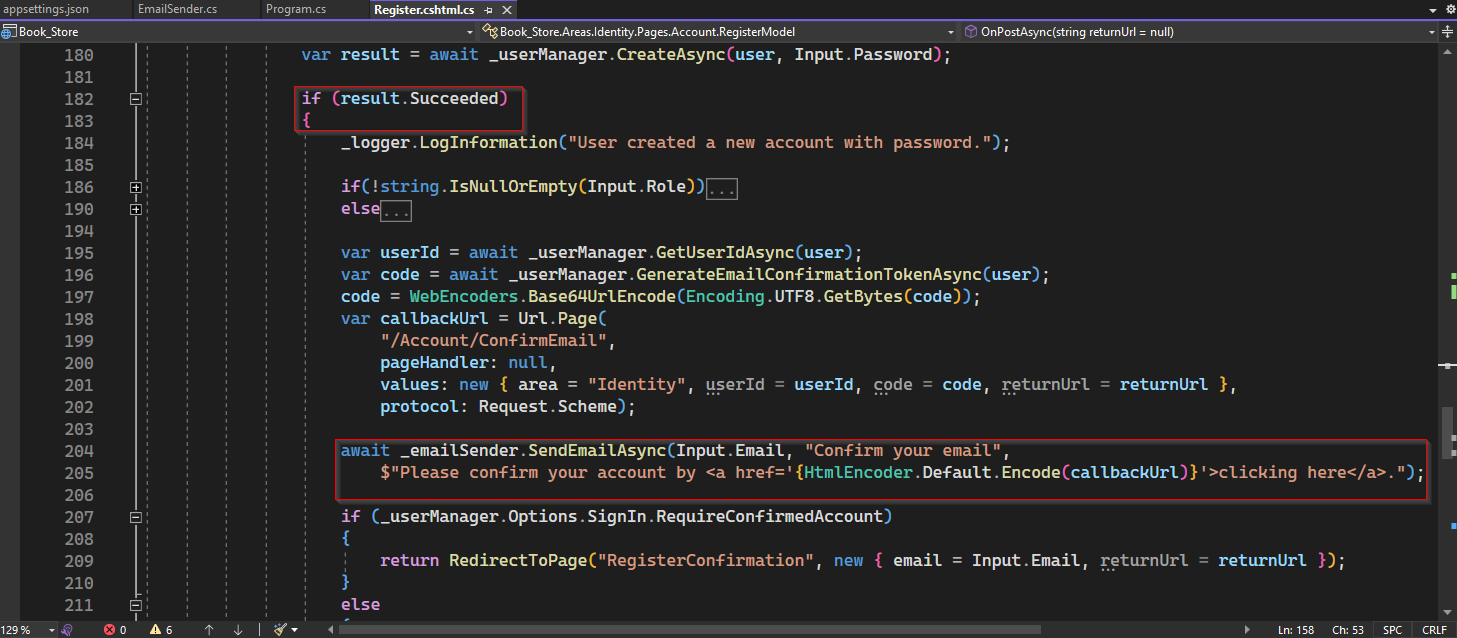


Next go to Program.cs and check if we have injected EmailSender correctly as below.

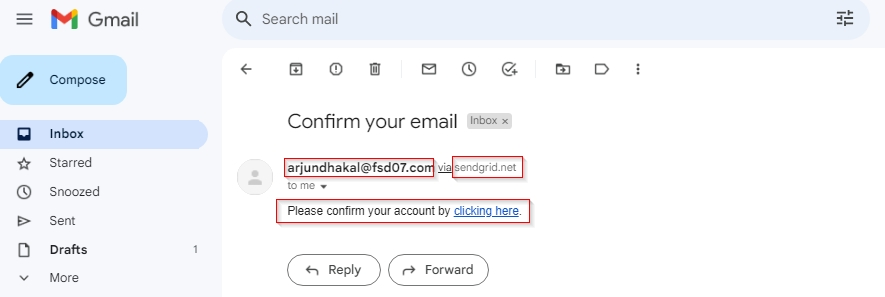


### 14.2.4 Use Email Service

On page model Register.cshtml.cs email service has been added by default on OnPostAsync method. When registration is successful, it automatically sends email to users email address to verify/confirm email address.



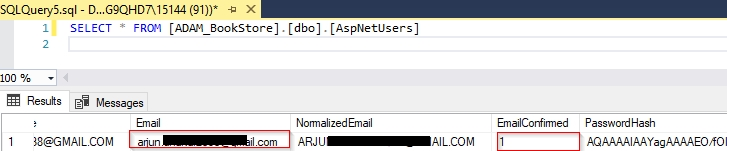
Now to test this functionality, lets register a new account and check if email is actually received.

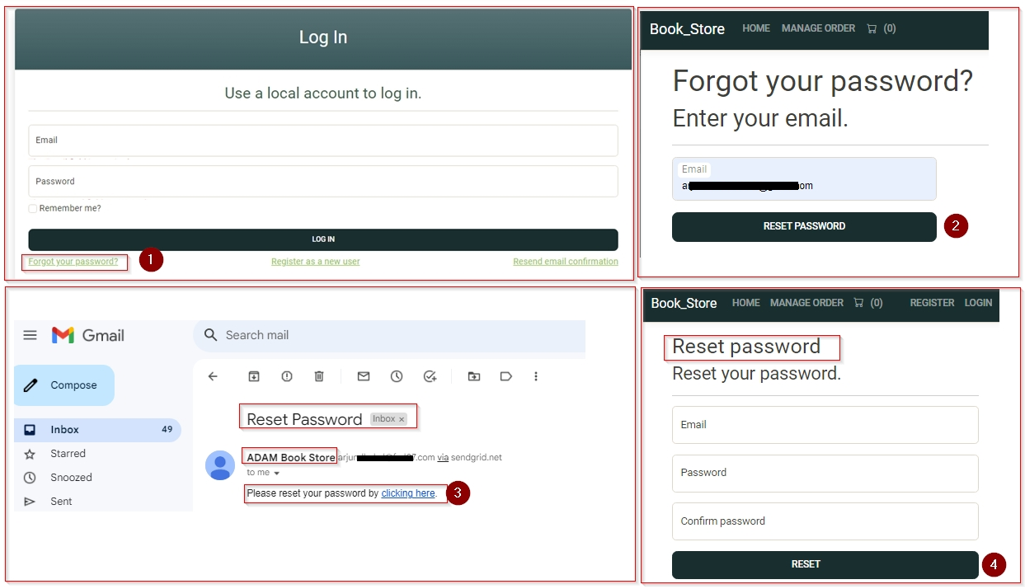
Once you successfully register, you will receive an email as follows.

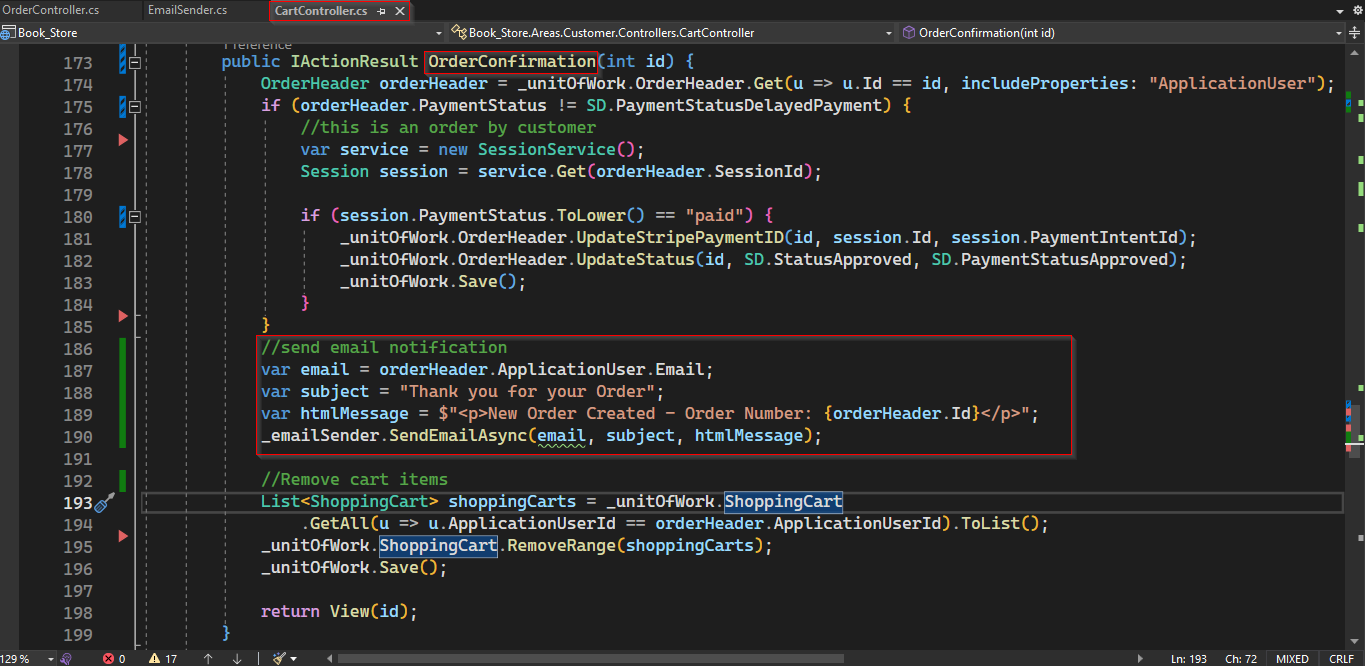
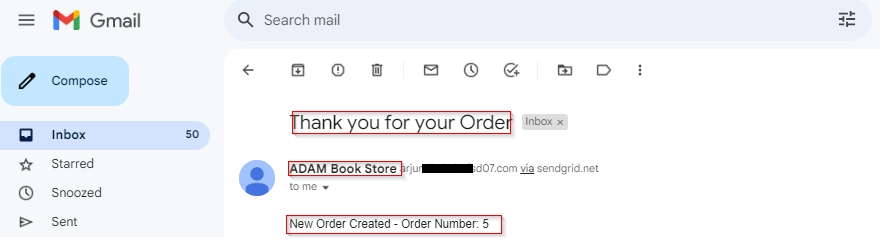
If you click link "clinking here", you will be redirected to the following.



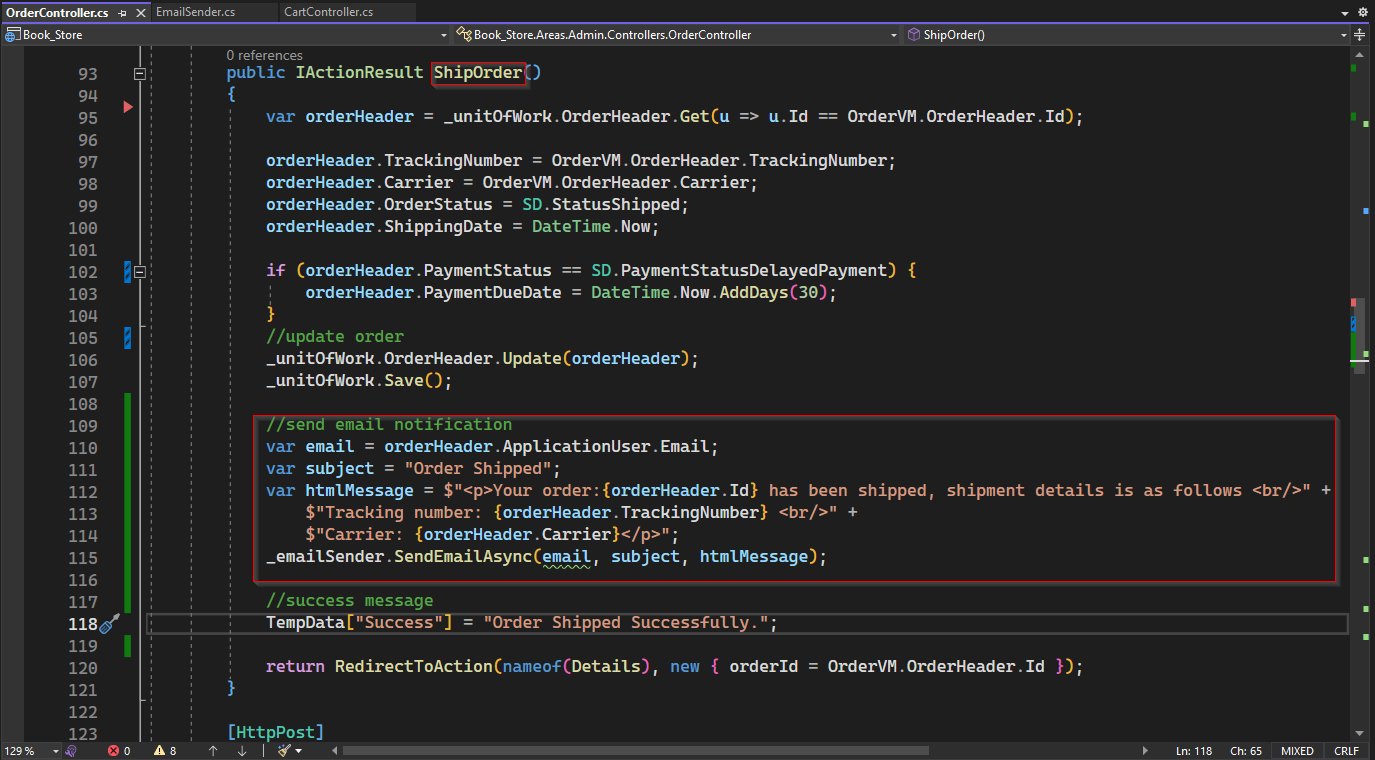
If you check Users table in database, you can see EmailConfirmed is 1 which means true.



Similarly, we send email to reset password. Note email to reset password will be sent if email has not been verified.

We can also send email notification on creating order by injecting IEmailSender on CartController and adding IEmailSender.EmailSenderAysnc method on OrderConfirmation as follows.Once order is place we get email notification as below.

Similarly, we can also send notification once order is shipped.



Once order is shipped, customer will receive email notification as follows.

