**Universitatea Tehnică a Moldovei**

**Lucrare de laborator 2**

**LA DISCIPLINA:** PROGRAMAREA IN RETEA

**CU TEMA:** SMTP | POP3 || IMAP

**Student:**

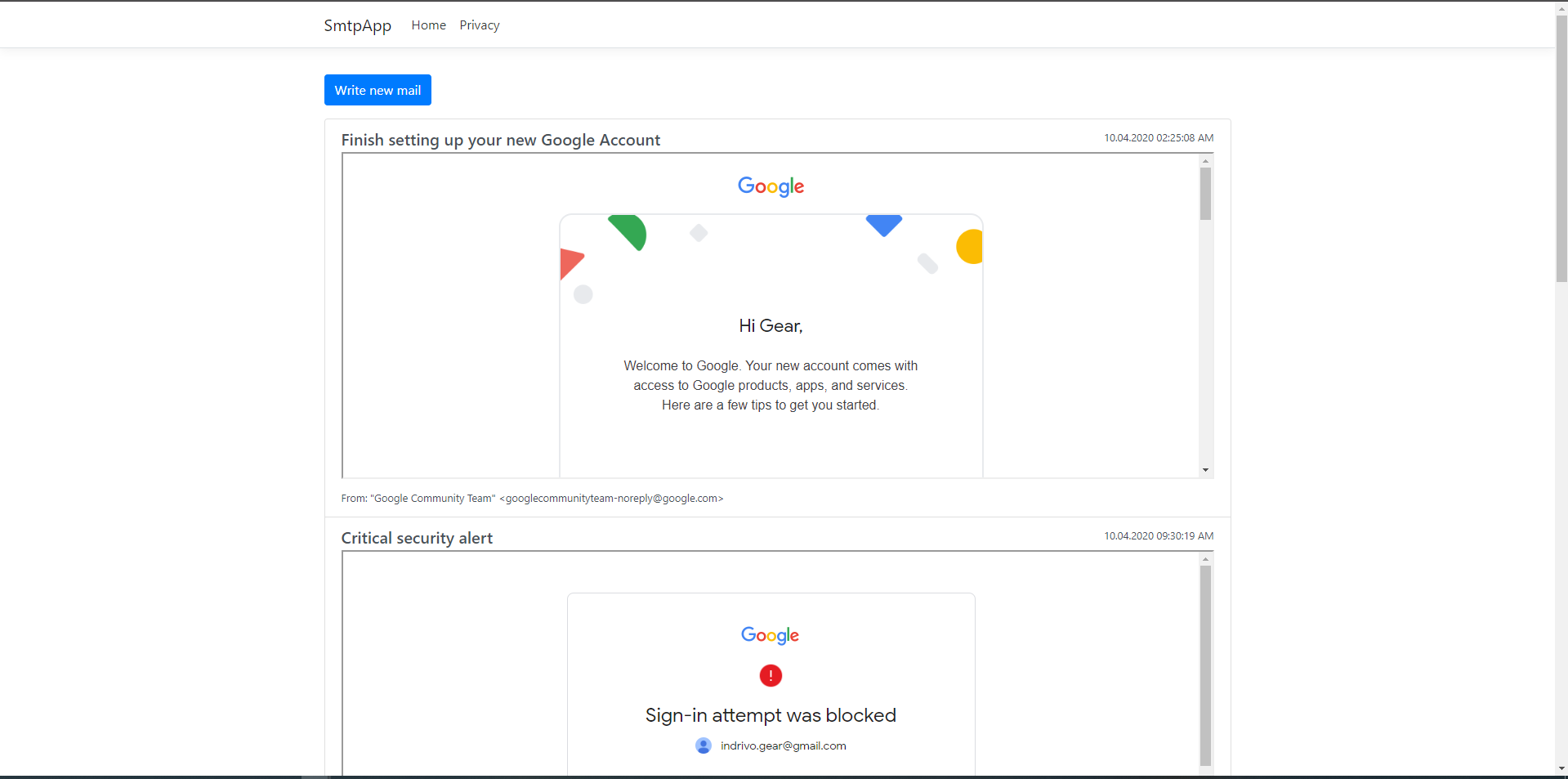
Lupei Nicolae FI-171

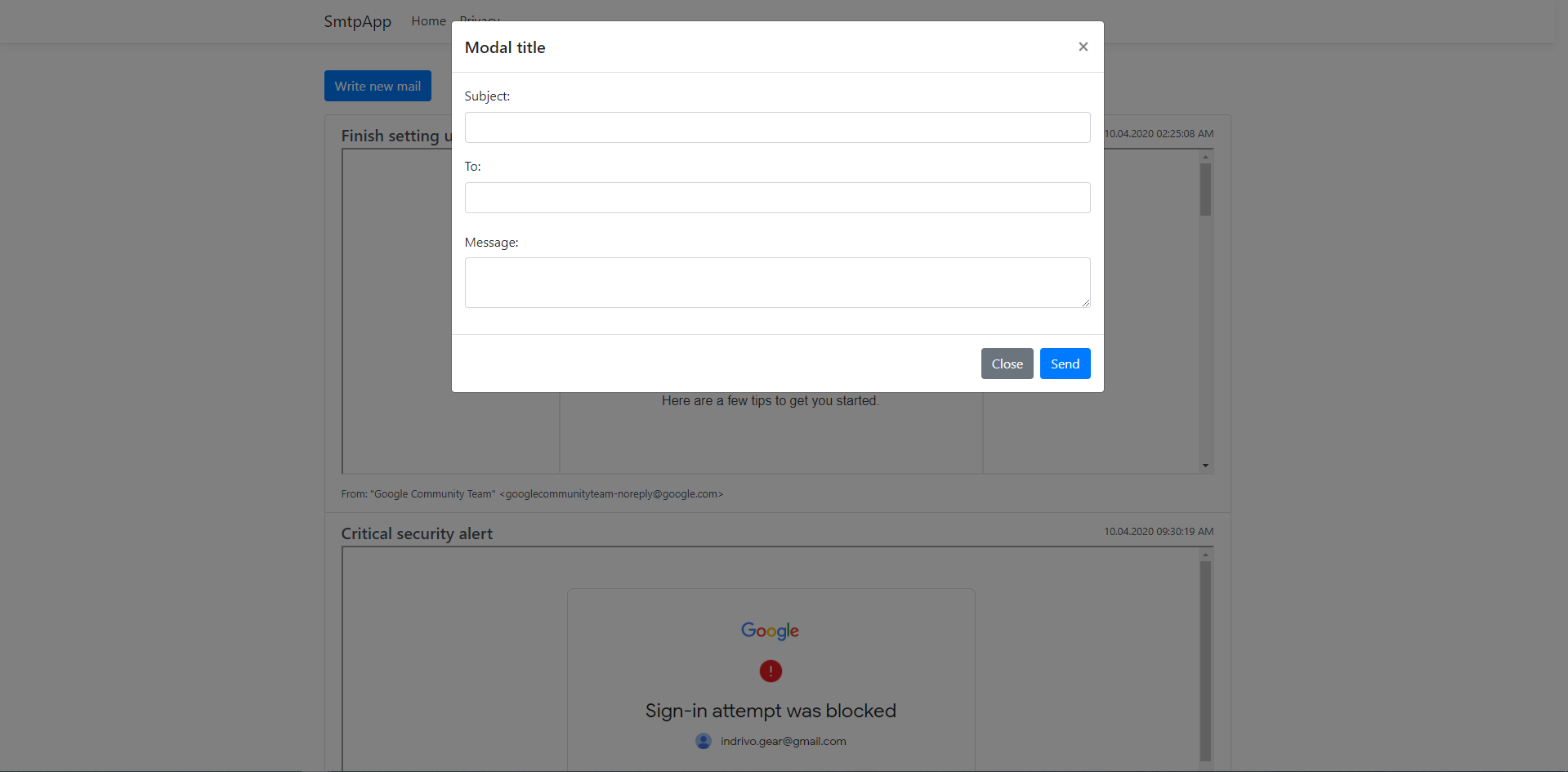
**Profesor:**

Boldumac Oleg

Chișinău – 2020

**Screens from sample:**





**Codul sursa:**

Implementare pentru SMTP client

using System;

using System.Net;

using System.Net.Mail;

using System.Text;

using System.Threading.Tasks;

using GR.Core.Abstractions;

using GR.Core.Extensions;

using Microsoft.AspNetCore.Identity.UI.Services;

using SmtpApp.ViewModels;

namespace SmtpApp.Services

{

    public class EmailSender : IEmailSender

    {

        /// <summary>

        /// Email settings

        /// </summary>

        private readonly IWritableOptions<SmtpSettingsViewModel> \_options;

        public EmailSender(IWritableOptions<SmtpSettingsViewModel> options)

        {

            if (options.Value == null) throw new Exception("Email settings not register in appsettings file");

            \_options = options;

        }

        /// <summary>

        /// Send email

        /// </summary>

        /// <param name="email"></param>

        /// <param name="subject"></param>

        /// <param name="htmlMessage"></param>

        /// <returns></returns>

        public async Task SendEmailAsync(string email, string subject, string htmlMessage)

        {

            if (!\_options.Value.Enabled || !email.IsValidEmail()) return;

            var settings = \_options.Value;

            try

            {

                using (var client = new SmtpClient())

                {

                    client.Port = settings.Port;

                    client.Host = settings.Host;

                    client.EnableSsl = settings.EnableSsl;

                    client.Timeout = settings.Timeout;

                    client.DeliveryMethod = SmtpDeliveryMethod.Network;

                    client.UseDefaultCredentials = false;

                    client.Credentials = new NetworkCredential(settings.NetworkCredential.Email, settings.NetworkCredential.Password);

                    var mailMessage = new MailMessage

                    {

                        BodyEncoding = Encoding.UTF8,

                        DeliveryNotificationOptions = DeliveryNotificationOptions.OnFailure,

                        From = new MailAddress(settings.NetworkCredential.Email),

                        Subject = subject,

                        Body = htmlMessage,

                        Priority = MailPriority.High,

                        IsBodyHtml = true

                    };

                    mailMessage.To.Add(email);

                    await client.SendMailAsync(mailMessage);

                }

            }

            catch (Exception e)

            {

                Console.WriteLine(e);

            }

        }

    }

}

Implementare pentru IMAP

using System;

using System.Collections.Generic;

using System.Threading.Tasks;

using GR.Core.Abstractions;

using MailKit;

using MailKit.Net.Imap;

using MailKit.Search;

using MimeKit;

using SmtpApp.Abstractions;

using SmtpApp.ViewModels;

namespace SmtpApp.Services

{

    public class EmailReader : IEmailReader

    {

        /// <summary>

        /// Imap settings

        /// </summary>

        private readonly ImapSettingsViewModel \_options;

        public EmailReader(IWritableOptions<ImapSettingsViewModel> options)

        {

            \_options = options.Value ?? throw new Exception("Email settings not register in appsettings file");

        }

        /// <summary>

        /// Get unread mails

        /// </summary>

        /// <returns></returns>

        public async Task<IEnumerable<MimeMessage>> GetUnreadMailsAsync()

        {

            var messages = new List<MimeMessage>();

            using var client = new ImapClient();

            await client.ConnectAsync(\_options.Host, \_options.Port, \_options.EnableSsl);

            // Note: since we don't have an OAuth2 token, disable

            // the XOAUTH2 authentication mechanism.

            client.AuthenticationMechanisms.Remove("XOAUTH2");

            await client.AuthenticateAsync(\_options.NetworkCredential.Email, \_options.NetworkCredential.Password);

            // The Inbox folder is always available on all IMAP servers...

            var inbox = client.Inbox;

            inbox.Open(FolderAccess.ReadOnly);

            var results = inbox.Search(SearchOptions.All, SearchQuery.Not(SearchQuery.Seen));

            foreach (var uniqueId in results.UniqueIds)

            {

                //Mark message as read

                //inbox.AddFlags(uniqueId, MessageFlags.Seen, true);

                var message = inbox.GetMessage(uniqueId);

                messages.Add(message);

            }

            await client.DisconnectAsync(true);

            return messages;

        }

        /// <summary>

        /// Get all emails

        /// </summary>

        /// <returns></returns>

        public async Task<IEnumerable<MimeMessage>> GetAllMailsAsync()

        {

            var messages = new List<MimeMessage>();

            using var client = new ImapClient();

            await client.ConnectAsync(\_options.Host, \_options.Port, \_options.EnableSsl);

            // Note: since we don't have an OAuth2 token, disable

            // the XOAUTH2 authentication mechanism.

            client.AuthenticationMechanisms.Remove("XOAUTH2");

            await client.AuthenticateAsync(\_options.NetworkCredential.Email, \_options.NetworkCredential.Password);

            // The Inbox folder is always available on all IMAP servers...

            var inbox = client.Inbox;

            inbox.Open(FolderAccess.ReadOnly);

            var results = await inbox.SearchAsync(SearchOptions.All, SearchQuery.All);

            foreach (var uniqueId in results.UniqueIds)

            {

                var message = inbox.GetMessage(uniqueId);

                //Mark message as read

                //inbox.AddFlags(uniqueId, MessageFlags.Seen, true);

                messages.Add(message);

            }

            await client.DisconnectAsync(true);

            return messages;

        }

    }

}

Pagina WEB

@using SmtpApp.Abstractions

@inject IEmailReader Reader

@{

    ViewData["Title"] = "Home Page";

    var messages = await Reader.GetAllMailsAsync();

}

<!-- Button trigger modal -->

<button type="button" class="btn btn-primary my-3" data-toggle="modal" data-target="#exampleModal">

    Write new mail

</button>

<div class="list-group">

    @foreach (var message in messages)

    {

        <a href="#" class="list-group-item list-group-item-action flex-column align-items-start">

            <div class="d-flex w-100 justify-content-between">

                <h5 class="mb-1">@message.Subject</h5>

                <small>@message.Date.ToString("dd'.'MM'.'yyyy hh:mm:ss tt")</small>

            </div>

            <p class="mb-1">

                <iframe height="400" width="100%" srcdoc="@message.HtmlBody"></iframe>

            </p>

            <small>From: @message.From.ToString()</small>

        </a>

    }

</div>

<!-- Modal -->

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

    <div class="modal-dialog modal-lg" role="document">

        <div class="modal-content">

            <div class="modal-header">

                <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>

                <button type="button" class="close" data-dismiss="modal" aria-label="Close">

                    <span aria-hidden="true">&times;</span>

                </button>

            </div>

            <div class="modal-body">

                <form id="sendMessageForm">

                    <div class="form-group">

                        <label for="Subject">Subject:</label>

                        <input type="text" name="Subject" Id="Subject" class="form-control" />

                    </div>

                    <div class="form-group">

                        <label for="To">To:</label>

                        <input type="text" name="To" Id="To" class="form-control" />

                    </div>

                    <div class="form-group">

                        <label for="Message" class="col-form-label">Message:</label>

                        <textarea class="form-control" id="Message" name="Message"></textarea>

                    </div>

                </form>

            </div>

            <div class="modal-footer">

                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

                <button id="sendData" type="button" class="btn btn-primary">Send</button>

            </div>

        </div>

    </div>

</div>

@section Scripts{

    <script>

$(document).ready(() => {

    $("#sendData").on("click", () => {

        const form = $("#sendMessageForm");

        const data = form.serializeArray();

        $.ajax({

            type: "post",

            url: "@Url.Action("SendEmail")",

            data: $.param(data)

        }).done((response) => {

            if (response.is\_success) {

                alert(response.result);

                $("#exampleModal").modal("hide");

            } else {

            }

        }).fail(e => {

        });

    });

});

    </script>

}

**Întrebări la apărarea laboratorului:**

1. **Scopul protocolului SMTP**

Simple Mail Transfer Protocol (prescurtat, SMTP; în traducere aproximativă Protocolul simplu de transfer al corespondenței) este un protocol simplu din suită de protocoale de Internet, care este folosit la transmiterea mesajelor în format electronic în rețea de calculatoare. SMTP folosește portul de aplicație 25 TCP („smtp”) și determină adresa unui server SMTP pe baza înregistrării MX (Mail eXchange, „schimb de corespodență”) din configurația serverului DNS.

Protocolul SMTP specifică modul în care mesajele de poștă electronică sunt transferate între procese SMTP aflate pe sisteme diferite. Procesul SMTP care are de transmis un mesaj este numit client SMTP iar procesul SMTP care primește mesajul este serverul SMTP. Protocolul nu se referă la modul în care mesajul ce trebuie transmis este trecut de la utilizator către clientul SMTP, sau cum mesajul recepționat de serverul SMTP este livrat utilizatorului destinatar și nici cum este memorat mesajul sau de câte ori clientul SMTP încearcă să transmită mesajul.

1. **Cum se poate verifica dacă serverul SMTP funcționează utilizînd linia de comandă?**

telnet smtp.servername.com 25

1. **Care sunt comenzile SMTP**

HELO

Este prima comandă SMTP: începe conversația identificând serverul expeditor și este, în general, urmată de numele său de domeniu.

EHLO

O comandă alternativă pentru a începe conversația, bazându-se pe faptul că serverul utilizează protocolul SMTP extins.

MAIL FROM

Cu această comandă SMTP, operațiunile încep: expeditorul precizează adresa de e-mail sursă în câmpul „From” și începe efectiv transferul de e-mailuri.

RCPT TO

Identifică destinatarul e-mailului; dacă există mai multe, comanda este pur și simplu repetată adresă prin adresă.

SIZE

Această comandă SMTP informează serverul de la distanță despre dimensiunea estimată (în termeni de octeți) a e-mailului atașat. De asemenea, poate fi utilizat pentru a raporta dimensiunea maximă a unui mesaj care trebuie acceptat de server.

DATA

CU comanda DATA conținutul de e-mail începe să fie transferat; este în general urmat de un cod de răspuns 354 dat de server, care dă permisiunea de a începe transmisia efectivă.

VRFY

Serverului i se cere să verifice dacă există o anumită adresă de e-mail sau nume de utilizator.

TURN

Această comandă este utilizată pentru a inversa rolurile între client și server, fără a fi necesară executarea unei noi cunoștințe.

AUTH

Cu comanda AUTH, clientul se autentifică pe server, dându-și numele de utilizator și parola. Este un alt strat de securitate pentru a garanta o transmisie adecvată.

RSET

Acesta comunică serverului că transmisia de e-mail curentă va fi încheiată, deși conversația SMTP nu va fi închisă (ca în cazul QUIT).

EXPN

Această comandă SMTP solicită o confirmare cu privire la identificarea listei de e-mailuri.

HELP

Este o solicitare a clientului pentru anumite informații care pot fi utile pentru transferul cu succes al e-mailului.

QUIT

Termină conversația SMTP.

1. **Pentru ce este nevoie de MUA, MSA, MTA și MDA**

MUA (agent utilizator de e-mail)

Aplicație client care permite primirea și trimiterea de e-mailuri. Poate fi o aplicație desktop cum ar fi Microsoft Outlook / Thunderbird / ... sau bazată pe web, cum ar fi Gmail / Hotmail / ... (aceasta din urmă se numește și Webmail).

MSA (Agent de trimitere prin poștă)

Un program de server care primește mail de la un MUA, verifică eventualele erori și îl transferă (cu SMTP) către MTA găzduit pe același server.

MTA (agent de transfer de poștă)

O aplicație server care primește mail de la MSA sau de la un alt MTA. Acesta va găsi (prin servere de nume și DNS) înregistrarea MX din zona DNS a domeniului destinatar pentru a ști cum să transferați poșta. Apoi transferă e-mailul (cu SMTP) către un alt MTA (care este cunoscut sub numele de retransmisie SMTP) sau, dacă a fost atins serverul destinatarului, către MDA.

Exemple de MTA sunt Postfix, Exim, Sendmail, qmail, ...

MDA (agent de livrare prin poștă)

Un program de server care primește e-mail de la serverul MTA și îl stochează în căsuța poștală. MDA este, de asemenea, cunoscut sub numele de LDA (Local Delivery Agent).

Un exemplu este Dovecot, care este în principal un server POP3 și IMAP care permite unui MUA să recupereze mail, dar include și un MDA care preia poștă de la un MTA și îl livrează la căsuța poștală a serverului.

1. **Care este diferența dintre porturile 25, 465 și 587?**

SMTP funcționează practic pe trei porturi -

Port 25 - port necriptat SMTP implicit

Port 2525 - este deschis pentru toate serverele SiteGround în cazul în care portul 25 este filtrat și trebuie să trimiteți e-mailuri necriptate cu SMTP

Port 465 - utilizat pentru a trimite mesaje folosind SMTP în siguranță

Port 25:

Portul SMTP 25 continuă să fie utilizat în principal pentru redarea SMTP. Transmiterea SMTP este transmiterea e-mailului de la serverul de e-mail la serverul de e-mail.

În majoritatea cazurilor, clienții de e-mail SMTP modern (Microsoft Outlook, Mail, Thunderbird etc.) nu ar trebui să utilizeze acest port. În mod tradițional, este blocat de ISP-urile rezidențiale și furnizorii de cloud hosting, pentru a reduce cantitatea de spam care este transmisă de pe computere sau servere compromise. Cu excepția cazului în care gestionați în mod specific un server de poștă, nu ar trebui să aveți trafic care să parcurgă acest port pe computer sau server. În caz contrar, veți observa o scădere a livrării prin e-mail și, în final, și o livrare redusă

Port 465:

IANA a alocat un nou serviciu acestui port și nu ar trebui să mai fie utilizat pentru comunicațiile SMTP.

Cu toate acestea, deoarece IANA a fost cândva recunoscut ca fiind valid, pot exista sisteme moștenitoare care sunt capabile să utilizeze doar această metodă de conectare. Drept urmare, Mailgun acceptă conexiuni SSL prin portul 465. De obicei, veți utiliza acest port numai dacă cererea dvs. o cere. O căutare rapidă pe Google și veți găsi numeroase articole ISP-uri ale furnizorilor de servicii inbox pentru consumatori care sugerează portul 465 ca configurație recomandată. Sperăm că acest lucru se va termina curând! Nu respectă RFC.

Port 587:

Acesta este portul implicit de trimitere prin poștă. Atunci când un client de e-mail sau un server de ieșire trimite un e-mail pentru a fi dirijat de un server de mail corespunzător, acesta ar trebui să folosească întotdeauna portul SMTP 587 ca port implicit.

ACEST PORT, ÎNSOȚIT DE CRIPTAREA TLS, VA ASIGURA TRANSMITEREA E-MAILULUI ÎN CONDIȚII DE SIGURANȚĂ ȘI URMÂND LINIILE DIRECTOARE STABILITE DE IETF.

Toți clienții Mailgun ar trebui să ia în considerare utilizarea portului 587 ca port SMTP implicit, cu excepția cazului în care sunteți blocat în mod explicit de rețeaua dvs. din amonte sau de furnizorul de găzduire.

1. **Care este diferența dintre porturile 110 și 995 ?**

POP3 funcționează practic pe două porturi -

Port 110 - implicit port ne-criptat POP3

Port 995 - acest port este necesar dacă doriți să vă conectați folosind POP3 în siguranță

1. **Care este diferența dintre porturile 143 și 993 ?**

IMAP funcționează pe două porturi -

Port 143 - este portul implicit necriptat IMAP

Port 993 - aveți nevoie de acest lucru pentru a vă conecta folosind IMAP în siguranță

1. **Cum funcționează protocolul SMTP ?**

SMTP adică Protocol de transfer de poștă simplă. Aceasta permite transmiterea mesajelor prin e-mail pe internet. Practic, protocolul gestionează e-mailurile care ies. Este folosit între servere pentru a trimite e-mailuri la un server și, de asemenea, folosit de servere pentru a împinge e-mailul către un alt server. Există și alte protocoale disponibile, dar SMTP este unul dintre cele mai comune protocoale între ele.

1. **Scopul protocoalelor POP3 și IMAP**

POP3 este utilizat pentru a primi e-mailuri de la server la client local. Când utilizați POP3 pentru a vă conecta contul de e-mail, e-mailurile sunt descărcate local pe sistemul dvs. și acestea sunt eliminate din serverul de e-mail. Prin aceasta, utilizatorii pot reduce spațiul de stocare al serverului lor web salvând mesajele pe computerul local.

IMAP este extins ca protocol de acces la mesaje Internet, utilizat pentru a primi mesaje de e-mail de pe internet. Prin protocolul IMAP, puteți primi e-mailurile dvs. pe mai multe sistem, deoarece mesajele primite vor rămâne pe serverul de e-mail.

1. **Diferența dintre POP3 și IMAP**

POP3 și IMAP sunt 2 metode diferite utilizate pentru a accesa e-mailurile de la un client sau software de la un terț. Fiecare metodă descarcă datele de e-mail de la AOL pe dispozitivul sau software-ul, cu toate acestea, acestea diferă în locul în care sunt stocate e-mailurile.

Examinați diferențele și alegeți protocolul potrivit pentru nevoile dvs. Indiferent dacă decideți POP3 sau IMAP pentru a vă accesa e-mailul, va trebui să configurați clientul de e-mail cu setările AOL .

IMAP (Internet Messaging Access Protocol)

• E-mailurile sunt stocate pe server.

• Mesajele trimise sunt stocate pe server.

• Mesajele pot fi sincronizate și accesate pe mai multe dispozitive.

POP3 (Protocolul poștal)

• E-mailurile sunt stocate pe un singur dispozitiv.

• Mesajele trimise sunt stocate pe un singur dispozitiv.

• E-mailurile pot fi accesate numai de la un singur dispozitiv.

• Dacă doriți să păstrați mesajele pe server, asigurați-vă că setarea „Keep email on server” este activată sau toate mesajele sunt șterse de pe server odată descărcate în aplicație sau software.

1. **Cum să verificați dacă există o adresă de e-mail fără a trimite un e-mail ?**

<https://blog.mailtrap.io/verify-email-address-without-sending/>

1. **Diferență dintre SSL și TLS**

Diferența între SSL și TLS.Termenii SSL și TLS sunt folosiți adesea în mod interdependent , fiind condiționate reciproc. De fapt, unul este predecesorul celuilalt. Protocolul SSL a fost întotdeauna folosit pentru a cripta și securiza transmisia datelor. De fiecare dată când o versiune nouă și mai sigură se lansează, numai numărul de versiune se modifică. De exemplu, există protocolul SSLv2.0. Cu timpul, el se actualizează și creează versiunea SSLv3.0. Pentru a nu genera utilizatorilor situații confuze între versiunile aceluiași protocol SSL, autorii au decis să redenumească a patra versiune în TLSv1.0. Cea mai recentă se dovedește a fi TLSv1.2. .