Step 0**. Django Settings:**

1. Settings.py

EMAIL\_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'

EMAIL\_HOST = 'smtp.office365.com'  # SMTP server for Microsoft 365

EMAIL\_PORT = 587  # Port for TLS

# Email credentials

EMAIL\_HOST\_USER = 'loginid@YourDomain.com'

EMAIL\_HOST\_PASSWORD = 'YourPassword# Your Microsoft 365 email password

# Security settings

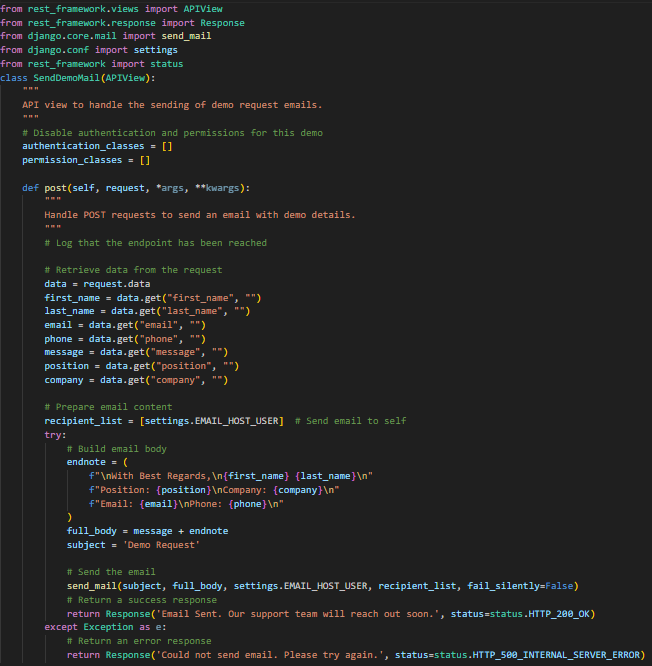
EMAIL\_USE\_TLS = True  # Use TLS for secure connection

EMAIL\_USE\_SSL = False  # SSL is not required for Microsoft 365

*Note: Gmail SMTP Settings Requires More Complex Process*

b)

views.py



Python Code:

from rest\_framework.views import APIView

from rest\_framework.response import Response

from django.core.mail import send\_mail

from django.conf import settings

from rest\_framework import status

class SendDemoMail(APIView):

    """

    API view to handle the sending of demo request emails.

    """

    # Disable authentication and permissions for this demo

    authentication\_classes = []

    permission\_classes = []

    def post(self, request, \*args, \*\*kwargs):

        """

        Handle POST requests to send an email with demo details.

        """

        # Log that the endpoint has been reached

        # Retrieve data from the request

        data = request.data

        first\_name = data.get("first\_name", "")

        last\_name = data.get("last\_name", "")

        email = data.get("email", "")

        phone = data.get("phone", "")

        message = data.get("message", "")

        position = data.get("position", "")

        company = data.get("company", "")

        # Prepare email content

        recipient\_list = [settings.EMAIL\_HOST\_USER]  # Send email to self

        try:

            # Build email body

            endnote = (

                f"\nWith Best Regards,\n{first\_name} {last\_name}\n"

                f"Position: {position}\nCompany: {company}\n"

                f"Email: {email}\nPhone: {phone}\n"

            )

            full\_body = message + endnote

            subject = 'Demo Request'

            # Send the email

            send\_mail(subject, full\_body, settings.EMAIL\_HOST\_USER, recipient\_list, fail\_silently=False)

            # Return a success response

            return Response('Email Sent. Our support team will reach out soon.', status=status.HTTP\_200\_OK)

        except Exception as e:

            # Return an error response

            return Response('Could not send email. Please try again.', status=status.HTTP\_500\_INTERNAL\_SERVER\_ERROR)

c) add url in urls.py

Step 1. **Enable authenticated client SMTP**

**Enable or disable authenticated client SMTP submission (SMTP AUTH) in Exchange Online**

* Article
* 01/23/2024
* 16 contributors

Feedback

**In this article**

1. [Enable SMTP AUTH for specific mailboxes](https://learn.microsoft.com/en-us/exchange/clients-and-mobile-in-exchange-online/authenticated-client-smtp-submission#enable-smtp-auth-for-specific-mailboxes)

SMTP client email submissions (also known as *authenticated SMTP submissions* or *SMTP AUTH*) are used in the following scenarios in Office 365 and Microsoft 365:

* POP3 and IMAP4 clients. These protocols only allow clients to *receive* email messages, so they need to use authenticated SMTP to *send* email messages.
* Applications, reporting servers, and multifunction devices that generate and send email messages.

The SMTP AUTH protocol is used for SMTP client email submissions, typically on TCP port 587. SMTP AUTH supports modern authentication (Modern Auth) through OAuth in addition to basic authentication. For more information, see [Authenticate an IMAP, POP or SMTP connection using OAuth](https://learn.microsoft.com/en-us/exchange/client-developer/legacy-protocols/how-to-authenticate-an-imap-pop-smtp-application-by-using-oauth).

Virtually all modern email clients that connect to Exchange Online mailboxes in Office 365 or Microsoft 365 (for example, Outlook, Outlook on the web, iOS Mail, Outlook for iOS and Android, etc.) don't use SMTP AUTH to send email messages.

Therefore, we highly recommend that you disable SMTP AUTH in your Exchange Online organization, and enable it only for the accounts (mailboxes) that still require it. There are two settings that can help you do this:

* An organization-wide setting to disable (or enable) SMTP AUTH.
* A per-mailbox setting that overrides the tenant-wide setting.

These settings only apply to mailboxes that are hosted in Exchange Online (Office 365 or Microsoft 365).

These settings only apply to mailboxes that are hosted in Exchange Online (Office 365 or Microsoft 365).

**Note**

* If *security defaults* is enabled in your organization, SMTP AUTH is already disabled in Exchange Online. To use SMTP AUTH, you need to disable *security defaults*. For more information, see [**Security defaults in Microsoft Entra ID**](https://learn.microsoft.com/en-us/entra/fundamentals/security-defaults)
* If your authentication policy disables basic authentication for SMTP, clients cannot use the SMTP AUTH protocol even if you enable the settings outlined in this article. For more information, see [**Disable Basic authentication in Exchange Online**](https://learn.microsoft.com/en-us/exchange/clients-and-mobile-in-exchange-online/disable-basic-authentication-in-exchange-online).

o resolve this issue, you can either disable security defaults or configure exceptions to allow SMTP authentication.

Step 2. **Enable authenticated client SMTP**

**Steps to Address the Security Defaults Lock**

**Option 1: Disable Security Defaults**

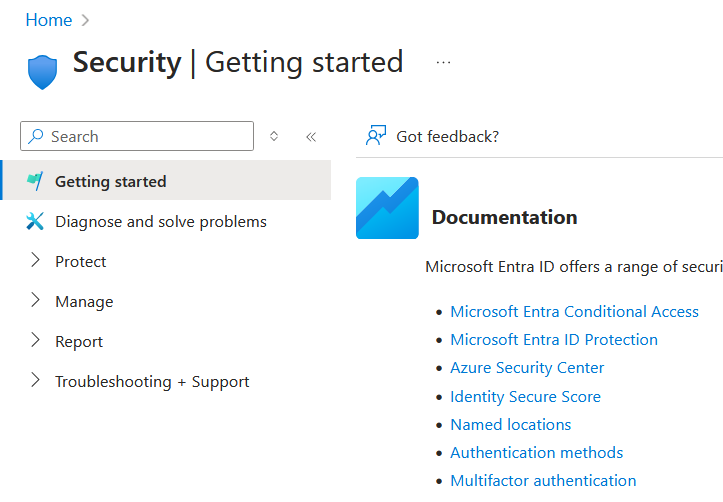
If you want to disable the security defaults (not recommended unless you have other security mechanisms like Conditional Access policies):

1. **Go to the Azure Active Directory Admin Center**:
   * Visit [Azure Portal](https://portal.azure.com/" \t "_new). ( <https://portal.azure.com/> )
   * Sign in with your admin account.
2. **Navigate to Azure Active Directory**:
   * On the left-hand side, click on **Azure Active Directory**.
3. **Find the Security Defaults Settings**:
   * In the **Azure AD** section, click on **Properties**.
   * Scroll down and find **Manage Security Defaults** at the bottom of the page.
4. **Disable Security Defaults**:
   * Click **Manage Security Defaults** and choose **No** to disable security defaults.
   * Click **Save**.

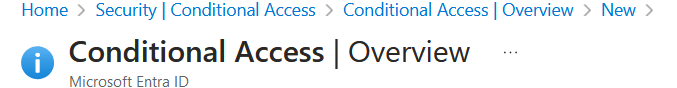
**Option 2: Use Conditional Access to Allow SMTP Authentication**

If security defaults are enforced and you prefer not to disable them for better protection, you can instead create a **Conditional Access policy** to allow SMTP authentication for certain users or apps.

1. **Go to the Azure Active Directory Admin Center**. ( <https://portal.azure.com/> )
2. **Navigate to ‘Azure Active Directory’ using a search > then select Security**.



1. Select **> Microsoft Entra Conditional Access**



1. **Create a new policy**:
   * Click on **+ New policy**.
   * Name your policy something like "Allow SMTP Authentication".
   * Under **Assignments**, select **Users and Groups** and choose the users that need SMTP access.
   * Under **Access controls**, select **Microsoft Exchange Online**.
   * In **Grant**, ensure that the policy grants access.
   * In **Sessions**, ensure that the policy grants access.
2. **Enable the Policy** and save it.

**Option 3: Contact Your Admin**

If you're not an admin or don't have access to make these changes, you'll need to contact your organization's administrator and request that they:

* Unlock your account.
* Review the **security defaults** settings.
* If necessary, follow one of the methods mentioned above to allow SMTP authentication.

After the lock is removed and SMTP authentication is allowed, your Django email setup should work correctly.