

SMTP-Authentifizierung für E-Mail-Benachrichtigungen

Aufgabe: E-Mail-Benachrichtigungen sollen über einen SMTP-Server verschickt werden. Welche Authentifizierungsverfahren unterstützt Accellence, welche Vor- und Nachteile haben sie und wie werden sie auf Seiten des SMTP-Clients und des SMTP-Servers eingerichtet?

Systemumgebung

- Anwendungen von Accellence mit SMTP-Client, z.B. EBÜS VA (Meldungen an den Systembetreuer, E-Mails an Kunden und Errichter, Weiterleitung von Bildern, Weiterleitung von Alarmen und Ereignissen), EBÜS Supervisor, EBÜS AutoPatrol, EBÜS Verbindungsnachweis, ...
- E-Mail-Server und -Anbieter: Microsoft 365 / Exchange Online, lokaler Microsoft Exchange Server, Google Workspace / Gmail, E-Mail-Provider (z.B. web.de, GMX, IONOS, Strato) sowie interne Relays/Smarthosts
- Voraussetzung: Software-Stand mit der Auswahl „Authentifizierung“ im SMTP-Client-Dialog (mind. 2.2.2.0). Beim ersten Start nach dem Update werden die bisherigen SMTP-Einstellungen automatisch übernommen.

Hintergrund

Die E-Mail-Benachrichtigungen aller Anwendungen nutzen denselben SMTP-Client. Bisher meldete er sich mit Benutzernamen und Passwort („Basic“) am Server an. Diese Anmeldung wird von immer mehr Servern abgelehnt:

- Microsoft 365 / Exchange Online:** Bis Ende Dezember 2026 bleibt das Verhalten unverändert. Danach ist SMTP AUTH Basic für bestehende Tenants standardmäßig deaktiviert (Administratoren können es weiter aktivieren); für neue Tenants steht es nicht mehr zur Verfügung. Ein endgültiges Abschaltdatum will Microsoft 2027 nennen (Message Center MC786329).
- Google (Gmail, Workspace):** Das Konto-Passwort wird für SMTP nicht mehr akzeptiert. Möglich sind App-Passwörter oder OAuth 2.0.
- Weitere Anbieter:** Meist weiterhin Benutzernamen/Passwort (häufig als App-Passwort), teils nur noch mit verschlüsselter Verbindung auf Port 465 oder 587.

Der in .NET enthaltene SMTP-Client unterstützt weder OAuth 2.0 noch verschlüsselte Verbindungen auf Port 465. Deshalb enthält unser SMTP-Client einen eigenen SMTP-Dialog; für Microsoft 365 steht zusätzlich der Versand über Microsoft Graph (HTTPS) zur Verfügung. Die Angaben zu Microsoft und Google entsprechen dem Stand vom Oktober 2026; bitte prüfen Sie bei der Einrichtung die aktuellen Hinweise der Hersteller.

Unterstützte Verfahren

Das Verfahren wird im SMTP-Client-Dialog der jeweiligen Funktion unter „Authentifizierung“ gewählt. Der Dialog zeigt nur die Eingabefelder an, die für das gewählte Verfahren nötig sind, und passt seine Größe automatisch an.

Verfahren	Typische Umgebung	Vorteile	Nachteile / Hinweise
Basic (User/Password)	E-Mail-Provider, lokale Mailserver; Microsoft 365 nur solange SMTP AUTH Basic erlaubt ist; Google nur mit App-Passwort	Einfach; funktioniert mit praktisch jedem Server; Port 587 (STARTTLS) und Port 465 (implizites TLS)	Passwort steht in der Konfiguration; wird durch MFA, Conditional Access oder Security Defaults blockiert; bei Microsoft 365 ab 12/2026 standardmäßig aus
OAuth 2.0 (Client Credentials)	Microsoft 365 / Exchange Online per SMTP	Kein Benutzerpasswort, keine MFA-Abfrage; Rechte pro Postfach vergebbar; gleicher Weg wie bisher (Port 587)	Einrichtung in Entra ID und Exchange PowerShell; Client-Secret läuft ab und muss erneuert werden; SMTP AUTH muss für das Postfach aktiv sein
Microsoft Graph (App-only)	Microsoft 365 / Exchange Online	Nur HTTPS (Port 443), proxyfähig; kein SMTP AUTH und kein Port 587 nötig; von Microsoft empfohlener Weg	Nur Microsoft 365; Nachrichten max. ca. 4 MB; Client-Secret läuft ab; Berechtigung Mail.Send sollte auf Postfächer eingeschränkt werden
OAuth 2.0 (Browser Login)	Google (Gmail, Workspace); Microsoft 365 mit Benutzerkonto	Für Gmail der moderne Weg ohne App-Passwort; kein Passwort gespeichert; Zugriff beim Anbieter widerrufbar	Einmalige Anmeldung im Browser nötig; hängt am Benutzerkonto (Passwortänderung, Richtlinien, Inaktivität); Google: App muss veröffentlicht sein
Windows (Integrated)	Lokaler Exchange Server in der Windows-Domäne	Keine Zugangsdaten in der Konfiguration; Anmeldung mit dem Windows-Konto der Anwendung	Nur in Domänenumgebungen; Konto der Anwendung braucht Postfach/Rechte; nur Verbindungen mit STARTTLS (nicht Port 465)
None (Relay)	Interne Relays und Smarthosts; Microsoft-365-Relay über Connector	Einfachste Konfiguration; keine Zugangsdaten	Keine Authentifizierung: der Server muss die Absenderadresse (IP) freigeben; feste IP-Adresse nötig; oft nur für eigene Empfänger-Domänen

Accellence Technologies GmbH • Garbsener Landstr. 10 • D-30419 Hannover

Tel. 0511 - 277.2400 • Fax 0511 - 277.2499 • E-Mail: info@accellence.de • Website: www.accellence.de

Entscheidungshilfe

Ihr E-Mail-Server	Empfohlenes Verfahren	Begründung
Microsoft 365 / Exchange Online	Microsoft Graph (App-only)	Zukunftssicher, nur Port 443, kein SMTP AUTH nötig
Microsoft 365, wenn nur SMTP erlaubt ist (z.B. Gerät/Software, die SMTP erwartet)	OAuth 2.0 (Client Credentials)	Wie bisher per SMTP (Port 587), aber ohne Passwort
Google Workspace / Gmail	OAuth 2.0 (Browser Login) oder Basic mit App-Passwort	Das Konto-Passwort wird nicht mehr akzeptiert
Lokaler Exchange Server	Windows (Integrated), alternativ Basic oder Relay	Keine Passwörter in der Konfiguration
E-Mail-Provider (web.de, GMX, IONOS, Strato ...)	Basic (ggf. App-Passwort), Port 587 oder 465	Üblicher Weg dieser Anbieter
Interner Relay / Smarthost	None (Relay)	Server vertraut der IP-Adresse des Absenders

Gemeinsame Einstellungen im SMTP-Client

Diese Felder gibt es bei (fast) allen Verfahren:

- **Adresse des SMTP-Servers** und **Port**: siehe Tabelle der Ports unten. Bei Microsoft Graph entfallen Server, Port und Verschlüsselung.
- **SSL/TLS**: Verschlüsselung einschalten (STARTTLS auf Port 587 und 25; auf Port 465 wird immer implizites TLS verwendet). Die Verfahren OAuth 2.0 senden Zugangsdaten nur verschlüsselt.
- **Auswahlfeld darunter (Verschlüsselungsprotokoll)**: „SystemDefault“ überlässt Windows die Wahl der sichersten gemeinsamen TLS-Version (empfohlen). SSL 3.0, TLS 1.0 und 1.1 sind unsicher und sollten nicht verwendet werden.
- **Akzeptiere alle Server-Zertifikate**: Nur für Tests verwenden! Sonst ist nicht sichergestellt, dass man mit dem richtigen Server spricht (siehe Abschnitt Zertifikate).
- **E-Mail-Adresse des Absenders / Empfängers**: Der Absender muss zum Konto passen (Postfach oder „Senden als“-Berechtigung), sonst lehnen viele Server die Nachricht ab.
- **Senden freigeben**: Erst wenn dieser Haken gesetzt ist, werden E-Mails versendet.
- **Verbindung zum Server prüfen** und **E-Mail senden**: Senden eine Test-E-Mail. Die Schaltfläche färbt sich gelb (in Arbeit), grün (erfolgreich) oder rot (Fehler); Details stehen im Statusfenster des Dialogs und im Logbuch.

Ports und Firewall

Die Freigaben werden nur ausgehend vom Rechner mit der Anwendung benötigt. Eingehende Freigaben auf Seiten unserer Anwendungen sind für unseren SMTP-Client nicht erforderlich.

Port	Ziel	Verwendung
587 TCP	SMTP-Server	SMTP Submission mit STARTTLS; empfohlen für Basic, OAuth 2.0 und Windows. Microsoft 365: smtp.office365.com. Gmail: smtp.gmail.com
465 TCP	SMTP-Server	SMTP mit implizitem TLS („SMTPS“); für Basic und Relay, nicht für Windows (Integrated)
25 TCP	SMTP-Server	Klassischer SMTP-Port, nur für Relay; wird von Providern und Firewalls oft gesperrt. Microsoft-365-Relay: <Ihre-Domäne>.mail.protection.outlook.com
443 TCP	login.microsoftonline.com, graph.microsoft.com	Microsoft: Token holen (alle OAuth-Verfahren) und Versand über Graph
443 TCP	accounts.google.com, oauth2.googleapis.com	Google: Anmeldung und Token (OAuth 2.0 mit Browser Login)
lokal	127.0.0.1 / localhost	Nur bei der Anmeldung im Browser: kurzzeitig wird ein zufälliger lokaler Port geöffnet. Keine Freigabe nach außen nötig.

Läuft der Zugriff ins Internet über einen Proxy oder eine Firewall mit TLS-Inspektion, muss der Proxy die genannten Ziele zulassen; bei TLS-Inspektion muss das Zertifikat der Firewall auf dem Rechner als vertrauenswürdig installiert sein.

Verfahren einrichten

1. Basic (Benutzername und Passwort)

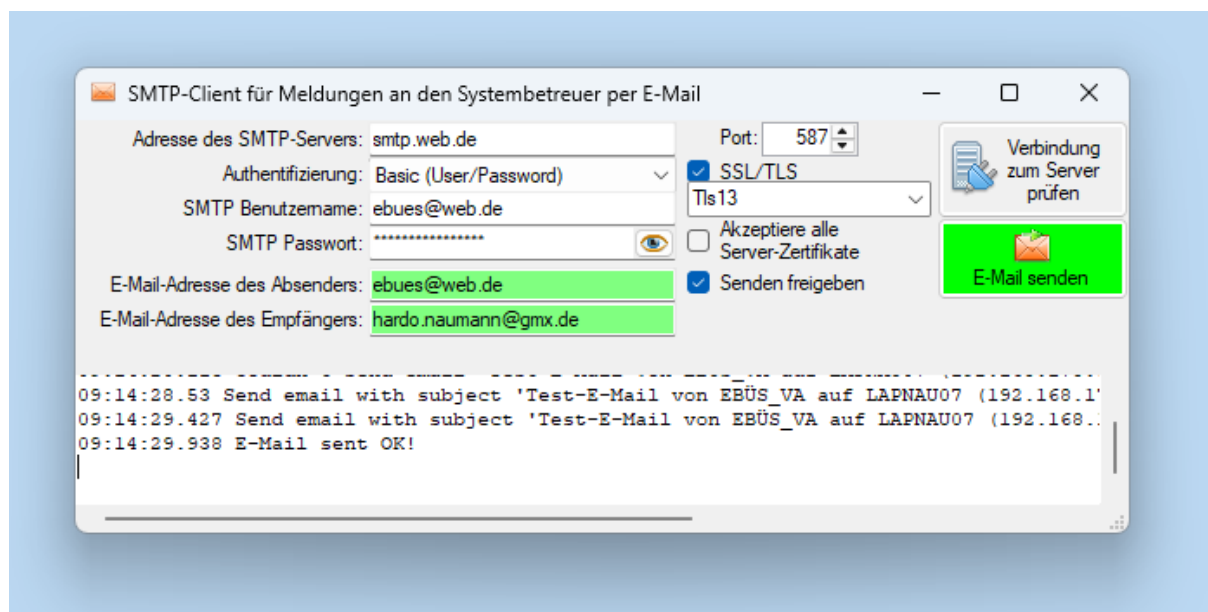
Auf Seiten des SMTP-Servers:

- Ein Postfach/Konto für den Versand anlegen (am besten ein eigenes Konto nur für die Benachrichtigungen mit minimalen Rechten).
- Bei Providern: SMTP-Zugriff für externe Programme im Kundenkonto aktivieren (z.B. „POP3/IMAP/SMTP-Zugriff erlauben“); bei Google und manchen Providern ein **App-Passwort** erzeugen (setzt Zwei-Faktor-Anmeldung voraus) und dieses statt des Konto-Passworts verwenden.
- Microsoft 365: Nur solange SMTP AUTH Basic erlaubt ist: im Postfach „Authentifiziertes SMTP“ aktivieren; MFA, Conditional Access und Security Defaults blockieren Basic.
- Port 587 (STARTTLS) oder 465 (implizites TLS) für ausgehende Verbindungen freigeben.

Im SMTP-Client:

1. Authentifizierung: **Basic (User/Password)**
2. Adresse des SMTP-Servers und Port eintragen (z.B. smtp.provider.de, 587 oder 465); SSL/TLS einschalten
3. SMTP Benutzername und SMTP Passwort (bzw. App-Passwort) eintragen
4. Absender- und Empfängeradresse eintragen, „Senden freigeben“ ankreuzen und mit „Verbindung zum Server prüfen“ testen

Beispiel einer SMTP-Konfiguration mit Benutzernamen und Passwort:



Das Verfahren hängt vom gewählten Port ab:

- Port **465**: Die Verbindung ist von Beginn an verschlüsselt („implizites TLS“). Der SMTP-Client verwendet dafür seinen eigenen SMTP-Dialog und meldet sich mit einem der Verfahren PLAIN, LOGIN oder CRAM-MD5 an, je nachdem, was der Server anbietet.
- Alle anderen Ports (z.B. **587** oder **25**): Die Verbindung beginnt unverschlüsselt und wird bei eingeschaltetem „SSL/TLS“ per STARTTLS auf Verschlüsselung umgestellt, bevor Benutzername und Passwort gesendet werden („explizites TLS“). Hier meldet sich der SMTP-Client mit Benutzername und Passwort an

2. OAuth 2.0 (Client Credentials) für Microsoft 365

Die Anwendung meldet sich nicht mit einem Benutzerpasswort, sondern als registrierte App mit Client-ID und Client-Secret an und sendet über SMTP (Port 587) im Namen eines Postfachs.

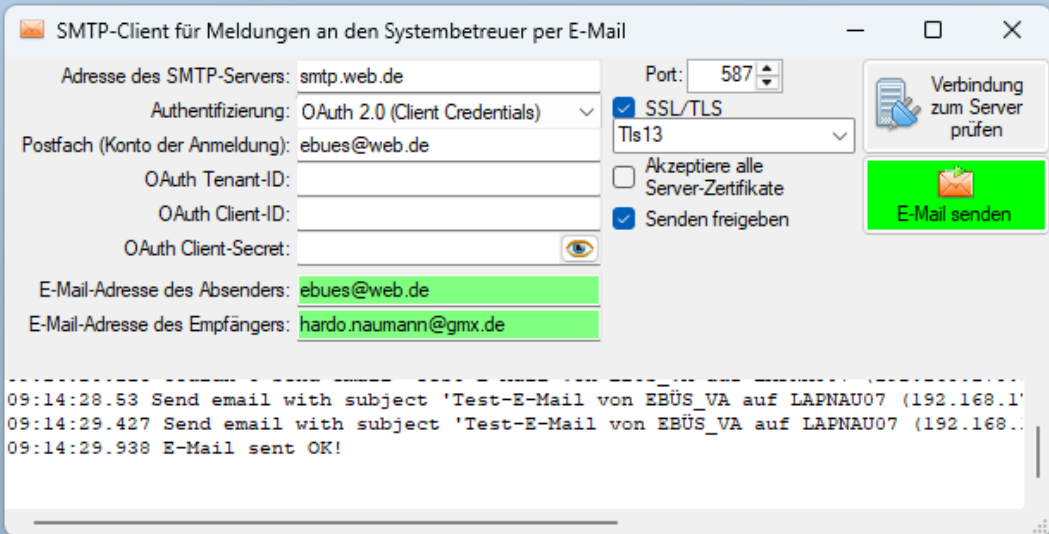
Auf Seiten von Microsoft 365 (einmalig, durch einen Administrator):

1. Im Entra-Admin-Center (entra.microsoft.com) unter „App-Registrierungen“ eine neue App registrieren (Name z.B. „EBÜS Mailversand“, nur dieses Verzeichnis). Notieren Sie **Anwendungs-(Client-)ID** und **Verzeichnis-(Mandanten-)ID**.
2. Unter „Zertifikate & Geheimnisse“ ein **Clientgeheimnis** anlegen. Notieren Sie sofort den **Wert** (nicht die Geheimnis-ID); er wird nur einmal angezeigt. Die Gültigkeit ist begrenzt: Erneuerung rechtzeitig einplanen.
3. Unter „API-Berechtigungen“ → „Berechtigung hinzufügen“ → „APIs, die meine Organisation verwendet“ → **Office 365 Exchange Online** → „Anwendungsberechtigungen“ → **SMTP.SendAsApp** hinzufügen und die **Administratorzustimmung** erteilen.
4. Dienstprinzipal in Exchange registrieren (Exchange Online PowerShell): `New-ServicePrincipal -AppId <Client-ID> -ObjectId <Objekt-ID der Unternehmensanwendung>`. Verwenden Sie hier die Objekt-ID der **Unternehmensanwendung** (Entra ID → Unternehmensanwendungen), nicht die der App-Registrierung.
5. Dem Dienstprinzipal Zugriff auf das Postfach geben: `Add-MailboxPermission -Identity "postfach@firma.de" -User <Objekt-ID der Unternehmensanwendung> -AccessRights FullAccess`. Falls Absender abweichen sollen, zusätzlich „Senden als“ vergeben (`Add-RecipientPermission`).
6. SMTP AUTH für das Postfach erlauben: `Set-CASMailbox -Identity "postfach@firma.de" -SmtpClientAuthenticationDisabled $false` (und ggf. auf Organisationsebene `Set-TransportConfig -SmtpClientAuthenticationDisabled $false`).
7. Die Berechtigungen können einige Minuten brauchen, bis sie wirksam werden. Bis dahin meldet der Server „535 5.7.3 Authentication unsuccessful“.

Im SMTP-Client:

1. Authentifizierung: **OAuth 2.0 (Client Credentials)**
2. Server: smtp.office365.com, Port: 587, SSL/TLS eingeschaltet
3. Postfach: SMTP-Adresse des Postfachs (Konto der Anmeldung); OAuth Tenant-ID, OAuth Client-ID und OAuth Client-Secret (Wert) eintragen
4. Absender: das Postfach (oder ein Alias mit „Senden als“-Recht); „Senden freigeben“ ankreuzen und testen

Unsere Bedienoberfläche zeigt die zum gewählten Authentifizierungsverfahren jeweils passenden Eingabefelder:



SMTP-Client für Meldungen an den Systembetreuer per E-Mail

Adresse des SMTP-Servers: smtp.web.de Port: 587

Authentifizierung: OAuth 2.0 (Client Credentials)

Postfach (Konto der Anmeldung): ebues@web.de

OAuth Tenant-ID:

OAuth Client-ID:

OAuth Client-Secret:

E-Mail-Adresse des Absenders: ebues@web.de

E-Mail-Adresse des Empfängers: hardo.naumann@gmx.de

☒ SSL/TLS

Tls13

☐ Akzeptiere alle Server-Zertifikate

☒ Senden freigeben

Verbindung zum Server prüfen

E-Mail senden

09:14:28.53 Send email with subject 'Test-E-Mail von EBÜS_VA auf LAPNAU07 (192.168.1...'

09:14:29.427 Send email with subject 'Test-E-Mail von EBÜS_VA auf LAPNAU07 (192.168...'

09:14:29.938 E-Mail sent OK!

3. Microsoft Graph (App-only) für Microsoft 365

Die E-Mail wird nicht per SMTP, sondern über die Microsoft-Graph-Schnittstelle (HTTPS, Port 443) versendet. Das ist der von Microsoft empfohlene Weg, braucht kein SMTP AUTH und keinen Port 587.

Auf Seiten von Microsoft 365 (einmalig, durch einen Administrator):

1. App registrieren und Clientgeheimnis anlegen wie in Verfahren 2 (Schritte 1 und 2).
2. Unter „API-Berechtigungen“ → **Microsoft Graph** → „Anwendungsberechtigungen“ → **Mail.Send** hinzufügen und die **Administratorzustimmung** erteilen.
3. **Wichtig:** Mail.Send erlaubt der App ohne weitere Einschränkung das Senden als jedes Postfach der Organisation. Beschränken Sie den Zugriff auf das Versandpostfach (z.B. mit einer Application Access Policy in Exchange Online: `New-ApplicationAccessPolicy -AppId <Client-ID> -PolicyScopeGroupId <Sicherheitsgruppe> -AccessRight RestrictAccess -Description "EBÜS";` Microsoft ersetzt diese Richtlinien schrittweise durch „RBAC for Applications“ – bitte aktuelle Hinweise beachten).
4. Das Versandpostfach muss ein Exchange-Online-Postfach mit Lizenz sein.

Im SMTP-Client:

1. Authentifizierung: **Microsoft Graph (App-only)** (Server, Port und Verschlüsselung entfallen)
2. Postfach: SMTP-Adresse des Versandpostfachs; OAuth Tenant-ID, OAuth Client-ID und OAuth Client-Secret (Wert) eintragen
3. Absender: das Postfach; „Senden freigeben“ ankreuzen und testen

Einschränkung: Eine Nachricht (einschließlich Anhängen) darf etwa 4 MB nicht überschreiten; Microsoft antwortet sonst mit „HTTP 413“. Anhänge und Bilder in E-Mails sind sonst ohne Einschränkung möglich.

4. OAuth 2.0 (Browser Login) für Google und Microsoft

Ein Anwender meldet sich einmal im Browser an und stimmt dem Versand zu. Die Anwendung erhält daraufhin ein „Refresh Token“ und holt sich damit bei Bedarf neue Zugriffstoken. Das Passwort des Kontos kennt die Anwendung nie. Das Refresh Token wird mit der übrigen SMTP-Konfiguration gespeichert und steht damit allen Instanzen zur Verfügung, die diese Konfiguration teilen.

Auf Seiten von Google (Gmail / Workspace):

1. In der Google Cloud Console ein Projekt anlegen und unter „APIs & Dienste“ → „OAuth-Zustimmungsbildschirm“ die App einrichten. **Veröffentlichungsstatus „In Produktion“** (bei Workspace: Nutzertyp „Intern“) wählen. Im Status „Test“ laufen die Refresh Token nach 7 Tagen ab!
2. Unter „Anmeldedaten“ eine **OAuth-Client-ID vom Typ „Desktop-App“** erstellen. Notieren Sie Client-ID und **Clientsschlüssel**.
3. Den Bereich (Scope) <https://mail.google.com/> für die App zulassen. Für das Konto muss SMTP-Zugriff erlaubt sein (bei Workspace durch den Administrator).

Auf Seiten von Microsoft 365 (mit Benutzerkonto statt App-only):

1. App registrieren; unter „Authentifizierung“ eine Plattform „Mobil- und Desktopanwendungen“ mit der Umleitungs-URI <http://localhost> hinzufügen und „Öffentliche Clientflows zulassen“ aktivieren.
2. Unter „API-Berechtigungen“ → Office 365 Exchange Online → „Delegierte Berechtigungen“ → **SMTP.Send** (und **offline_access**) hinzufügen; ggf. Administratorzustimmung erteilen.
3. Für das Postfach muss SMTP AUTH erlaubt sein (siehe Verfahren 2, Schritt 6). Ein Clientgeheimnis ist nicht nötig.

Im SMTP-Client:

1. Authentifizierung: **OAuth 2.0 (Browser Login)**
2. OAuth Anbieter: **Microsoft** oder **Google** wählen (Server und Port werden mit smtp.office365.com bzw. smtp.gmail.com, Port 587 vorbelegt, wenn dort noch nichts eingetragen war)
3. Postfach: E-Mail-Adresse des Kontos, mit dem die Anmeldung erfolgt; OAuth Client-ID eintragen (Microsoft zusätzlich ggf. die Tenant-ID; leer = alle Konten), bei Google auch das Client-Secret
4. Auf **Anmelden...** klicken: Der Standardbrowser öffnet sich; anmelden und den Zugriff erlauben. Danach zeigt der Dialog „Angemeldet“.
5. „Senden freigeben“ ankreuzen und mit „Verbindung zum Server prüfen“ testen

Mit **Abmelden** wird das gespeicherte Refresh Token gelöscht. Es wird auch gelöscht, wenn Anbieter, Mandant oder Client-ID geändert werden; dann ist eine neue Anmeldung nötig. Die Anmeldung muss an einem Rechner mit Browser durchgeführt werden; sie ist nach 3 Minuten ohne Abschluss abgelaufen.

5. Windows (Integrated) für lokale Exchange Server

Die Anwendung meldet sich mit dem Windows-Konto an, unter dem sie läuft (Kerberos/NTLM). In der Konfiguration sind keine Zugangsdaten nötig.

Auf Seiten des Exchange Servers:

- Am Empfangsconnector (z.B. „Client Frontend“, Port 587) muss als Authentifizierungsmechanismus „**Integrierte Windows-Authentifizierung**“ zugelassen sein und TLS (STARTTLS) angeboten werden.
- Das Windows-Konto, unter dem die Anwendung läuft, benötigt ein Postfach und darf in dessen Namen senden. Läuft die Anwendung als Dienst, muss das Dienstkonto ein Domänenkonto sein.
- Der Servername in der Konfiguration muss zum Namen im TLS-Zertifikat des Servers passen (Kerberos benötigt außerdem einen passenden Dienstprinzipalnamen).

Im SMTP-Client:

1. Authentifizierung: **Windows (Integrated)**
2. Server (FQDN des Exchange Servers) und Port 587 (oder 25) eintragen, SSL/TLS einschalten; Benutzernamen und Passwort entfallen
3. Absender- und Empfängeradresse eintragen, „Senden freigeben“ ankreuzen und testen

Hinweis: Mit Port 465 (implizites TLS) ist dieses Verfahren nicht möglich; der SMTP-Client meldet das als Fehler.

6. None (Relay ohne Anmeldung)

Der Server nimmt die E-Mail ohne Anmeldung an, weil er der Absender-IP-Adresse vertraut. Das ist nur in kontrollierten Netzen vertretbar.

Auf Seiten des Servers:

- Exchange Server: einen **Empfangsconnector (Relay)** anlegen, der die IP-Adresse des Rechners mit der Anwendung zulässt; nur die tatsächlich benötigten Absender-IP-Adressen eintragen und das Relay möglichst auf eigene Empfänger-Domänen beschränken.
- Microsoft 365: Ein **Connector** (Partner/Ihre Organisation) bindet den Versand an Ihre **feste öffentliche IP-Adresse**; versendet wird an <Ihre-Domäne>.mail.protection.outlook.com auf Port 25.
- Unabhängig vom Server: SPF-Eintrag der Absenderdomäne beachten, sonst werden die E-Mails ggf. als Spam eingestuft.

Im SMTP-Client: Authentifizierung **None (Relay)**; Server und Port (25, 587 oder 465); SSL/TLS nach Möglichkeit einschalten (Port 465: immer verschlüsselt); Absender, Empfänger, „Senden freigeben“.

Zertifikate und Verschlüsselung

- Der SMTP-Client prüft das Zertifikat des Servers wie ein Webbrowser: Es muss von einer Zertifizierungsstelle stammen, der der Rechner vertraut, noch gültig sein und zum eingetragenen Servernamen passen.
- Bei einer internen Zertifizierungsstelle muss deren Stammzertifikat im Windows-Zertifikatsspeicher „Vertrauenswürdige Stammzertifizierungsstellen“ des Rechners installiert sein. Siehe auch den Sicherheitshinweis www.ebues.de/KB/000016.
- Tragen Sie den Servernamen so ein, wie er im Zertifikat steht (nicht die IP-Adresse).
- Das Häkchen „Akzeptiere alle Server-Zertifikate“ deaktiviert diese Prüfung. Verwenden Sie es nur vorübergehend zum Testen, da sonst Zugangsdaten an einen falschen Server gelangen könnten.
- Microsoft 365 und Google verwenden Zertifikate öffentlicher Zertifizierungsstellen; hier ist keine zusätzliche Installation nötig, solange die Windows-Zertifikate des Rechners aktuell sind (Windows Update).

Sicherheit und Betrieb

- **Zugangsdaten schützen:** Passwörter, Client-Secrets und Refresh Token stehen in der Konfiguration der Anwendung. Schützen Sie die Konfigurationsdateien durch Dateirechte; in den Dialogen werden Passwortfelder nur für Benutzer mit dem Recht zum Anzeigen vertraulicher Konfigurationsdaten sichtbar.
- **Eigenes Versandkonto:** Verwenden Sie ein eigenes Postfach/Konto für Benachrichtigungen mit minimalen Rechten statt eines persönlichen Kontos.
- **Ablauf beachten:** Client-Secrets haben ein Ablaufdatum (Microsoft: höchstens 24 Monate). Tragen Sie das Datum in Ihren Kalender ein und erneuern Sie das Secret rechtzeitig, sonst bleiben die Benachrichtigungen aus. Dasselbe gilt für Passwörter und App-Passwörter, die geändert werden.
- **Sendebegrenzungen:** Anbieter begrenzen die Menge (z.B. Exchange Online: 30 Nachrichten pro Minute und 10.000 Empfänger pro Tag je Postfach). Die Anwendungen bündeln Meldungen, bei Massenergebnissen kann es dennoch zu Verzögerungen kommen.
- **Überwachen:** Testen Sie den Versand nach jeder Änderung und gelegentlich im laufenden Betrieb mit „E-Mail senden“. Fehler werden im Statusfenster des Dialogs und im Logbuch protokolliert.

Was tun wenn ...

Meldung im Statusfenster	Mögliche Ursache	Abhilfe
SMTP AUTH XOAUTH2 failed: 535 5.7.3 Authentication unsuccessful	Dienstprinzipal oder Postfach-berechtigung fehlt bzw. noch nicht wirksam; SMTP AUTH für das Postfach aus; falsche Objekt-ID	Verfahren 2 Schritte 4–6 prüfen; einige Minuten warten
OAuth 2.0 token request failed (HTTP 400/401) ... AADSTS7000215 Invalid client secret	Secret falsch (Geheimnis-ID statt Wert) oder abgelaufen	Neues Secret anlegen und den Wert eintragen
... AADSTS700016 Application ... not found	Client-ID oder Tenant-ID falsch	Werte aus der App-Registrierung prüfen
... AADSTS65001 / consent	Administratorzustimmung fehlt	Zustimmung in den API-Berechtigungen erteilen
Microsoft Graph sendMail failed (HTTP 403) ErrorAccessDenied	Mail.Send nicht erteilt oder Application Access Policy sperrt das Postfach	Berechtigung/Policy prüfen
Microsoft Graph sendMail failed (HTTP 404)	Postfach existiert nicht oder hat keine Exchange-Online-Lizenz	Postfachadresse prüfen
Microsoft Graph sendMail failed (HTTP 413)	Nachricht größer als ca. 4 MB	Anhänge verkleinern
SMTP server ... does not offer STARTTLS	Falscher Port oder Server ohne Verschlüsselung	Port 587 bzw. 465 verwenden
SMTP server ... does not offer AUTH XOAUTH2	Server unterstützt OAuth 2.0 nicht	Servername/Port prüfen (z.B. smtp.office365.com:587)
SMTP AUTH PLAIN/LOGIN failed: 535	Benutzername oder Passwort falsch; bei Google/Providern App-Passwort nötig	Zugangsdaten bzw. App-Passwort prüfen
... timed out	Port durch Firewall/Proxy gesperrt	Ausgehende Freigabe für Port und Ziel prüfen
Zertifikatsfehler z.B. „The remote certificate is invalid“	Zertifikat nicht vertrauenswürdig, abgelaufen oder Servername passt nicht	Abschnitt Zertifikate; nur zum Test: „Akzeptiere alle Server-Zertifikate“
OAuth 2.0 login is missing, please log in first	Beim Browser Login wurde noch nicht angemeldet oder die Anmeldung wurde gelöscht	Auf „Anmelden...“ klicken
OAuth login failed: access_denied / invalid_grant	Zugriff im Browser abgelehnt; Anmeldung abgelaufen oder widerrufen (z.B. Google im Testmodus nach 7 Tagen)	Neu anmelden; Google-App auf „In Produktion“ stellen

Weitere Informationen

- Microsoft: OAuth für IMAP, POP und SMTP – learn.microsoft.com/exchange/client-developer/legacy-protocols/how-to-authenticate-an-imap-pop-smtp-application-by-using-oauth
- Microsoft Graph: E-Mails mit MIME-Inhalt senden – learn.microsoft.com/graph/outlook-send-mime-message
- Google: OAuth 2.0 für SMTP (XOAUTH2) – developers.google.com/workspace/gmail/imap/xoauth2-protocol
- Weiterleitung von Ereignissen per E-Mail: www.ebues.de/KB/000021

Lassen Sie uns bitte gern gemeinsam beraten, welches Verfahren für Ihre E-Mail-Umgebung am besten passt. Dies hängt u.a. von den Sicherheits-Richtlinien ab, die in Ihrem Haus gelten.

Stand: 11.10.2026, Dipl.-Ing. Hardo Naumann

AccKB... steht für die Knowledge Base (Wissensdatenbank) von accellence
Wir stellen Ihnen [hier](#) kostenlos und unverbindlich nützliche Informationen zu vielen Themen bereit