

MS365 App-Only-Zugriff für mailclient4mia einrichten

Setup-Anleitung für den Tenant-Administrator

Diese Anleitung richtet sich an die Person, die in Ihrer Organisation Administratorrechte für Microsoft 365 / Entra ID besitzt. Sie richten damit einen Zugriff ein, mit dem der Mailclient E-Mails über das Postfach senden und abrufen kann, ohne dass sich danach jemand wiederkehrend im Browser anmelden muss.

Dauer: ca. 45–60 Minuten aktive Arbeit – plus mehrere Wartephasen von bis zu 60 Minuten (teils länger) pro Änderung, bis Microsoft-seitige Einstellungen tatsächlich wirksam werden. Planen Sie insgesamt eher einen halben bis ganzen Tag ein, nicht nur die reine Klickzeit.

Benötigte Rechte: Globaler Administrator oder Anwendungsadministrator (für Entra ID) UND Exchange-Administrator (für die PowerShell-Schritte in Exchange Online – das ist neu gegenüber der ersten Version dieser Anleitung).

Werkzeuge: Browser (Entra Admin Center) sowie Exchange-Online-PowerShell für mehrere Schritte, nicht mehr nur für den letzten.

***Hinweis:** Diese Version der Anleitung wurde nach einer mehrstündigen Live-Fehlersuche in einer neu angelegten Test-Umgebung überarbeitet. Drei Schritte fehlten in der ersten Fassung komplett (Schritt 5, 6 und 9 unten) – ohne sie schlägt die Authentifizierung trotz korrekt aussehender Konfiguration fehl. Der Anhang „Fehlerbehebung“ am Ende fasst die typischen Fehlermeldungen und ihre Ursache zusammen.*

Inhaltsverzeichnis

- Warum diese Umstellung?
- Schritt 1: App in Entra ID registrieren
- Schritt 2: Application Permissions vergeben
- Schritt 3: Admin Consent erteilen
- Schritt 4: Client Secret erzeugen
- Schritt 5: Service Principal in Exchange registrieren (NEU)
- Schritt 6: Postfach-Berechtigungen für den Service Principal setzen (NEU)
- Schritt 7: Werte an den Mailclient-Betreiber übergeben
- Schritt 8: Zugriff auf die richtigen Postfächer beschränken
- Schritt 9: Authentifiziertes SMTP tenant-weit aktivieren (NEU)
- Schritt 10: Protokoll-Freischaltung und Lizenz prüfen (NEU)
- Verifikation
- Anhang: Fehlerbehebung – Fehlermeldung → Ursache → Lösung

Warum diese Umstellung?

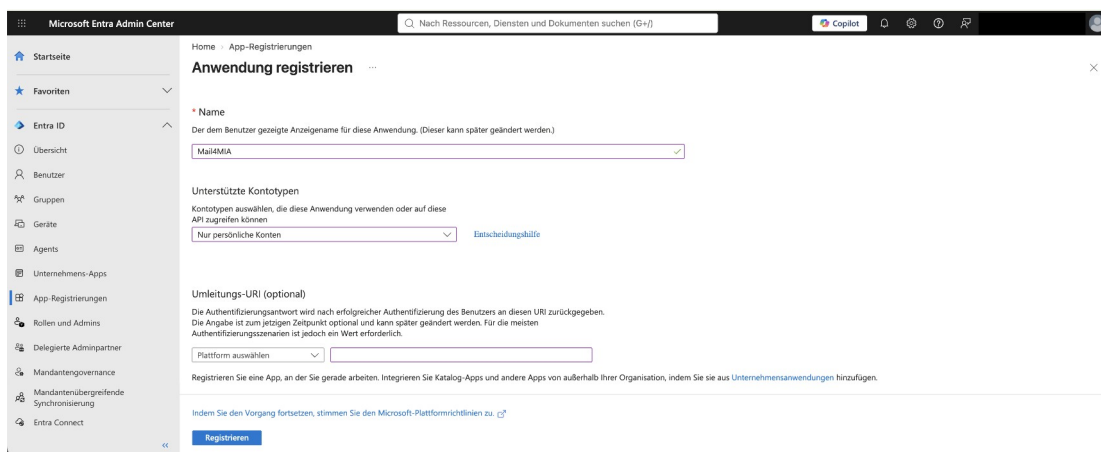
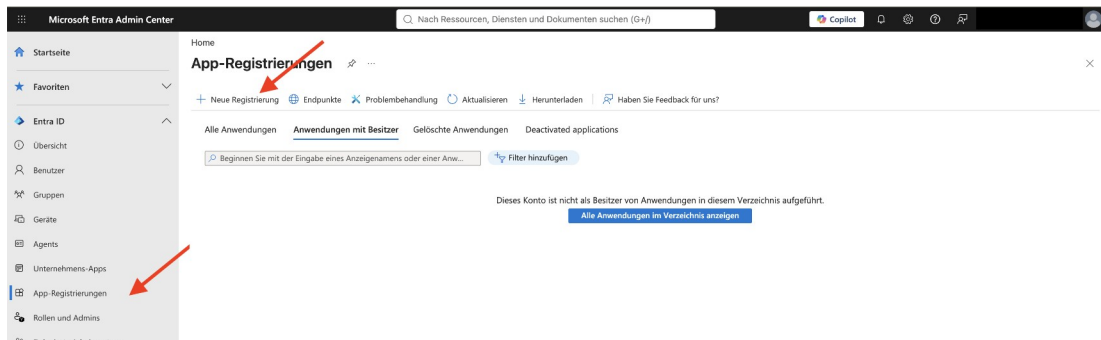
Bisher musste sich nach der Einrichtung regelmäßig eine Person im Browser unter microsoft.com/devicelogin anmelden, damit der Mailclient weiter auf das Postfach zugreifen darf. Mit dem OAuth2 Client-Credentials-Flow meldet sich stattdessen die Anwendung selbst bei Microsoft an – mit einer eigenen Kennung statt mit einem Benutzerkonto. Es ist danach keine Person mehr im Anmeldevorgang beteiligt.

Technischer Hintergrund: Microsoft Learn – Client Credentials Grant Flow. Der Mailclient ist von außen nicht erreichbar; ein Rückruf (Redirect/Callback) an ihn ist daher ohnehin nie möglich. Der Client-Credentials-Flow braucht dafür aber auch keinen Rückruf – er passt hier besonders gut.

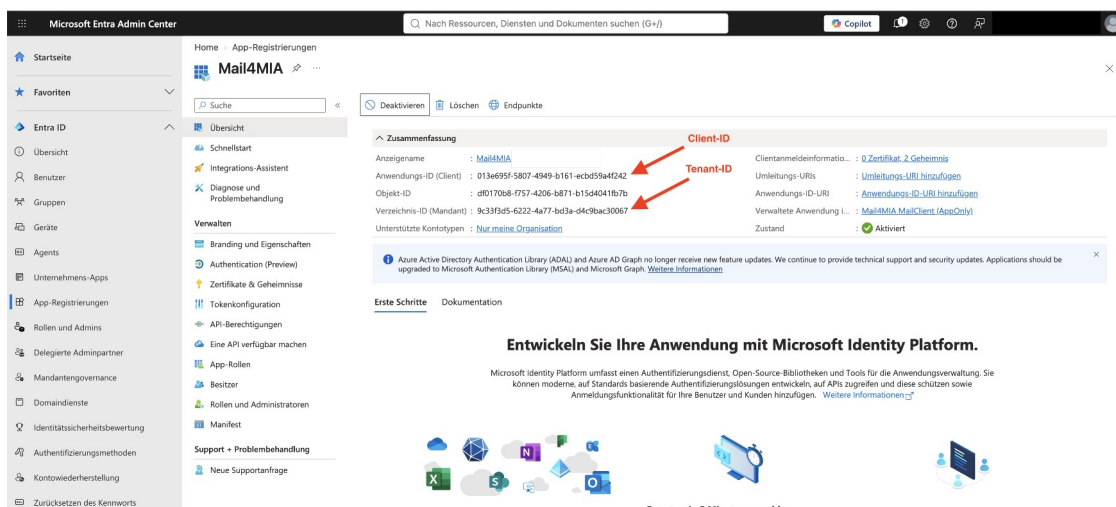
Schritt 1: App in Entra ID registrieren

Legen Sie eine eigene, ausschließlich für diesen Zweck genutzte App-Registrierung an.

1. Öffnen Sie das [Microsoft Entra Admin Center](#) → Entra ID → App-Registrierungen → Neue Registrierung.
2. Vergeben Sie einen sprechenden Namen, z. B. mailclient4mia – App-Only.
3. Unterstützte Kontotypen: „Nur Konten in diesem Organisationsverzeichnis“ (Single Tenant).
4. Redirect-URI: leer lassen – der Client-Credentials-Flow benötigt keine.



Notieren Sie sich von der Übersichtsseite der neuen App direkt Anwendungs-ID (Client) und Verzeichnis-ID (Mandant/Tenant) – beide werden später benötigt.

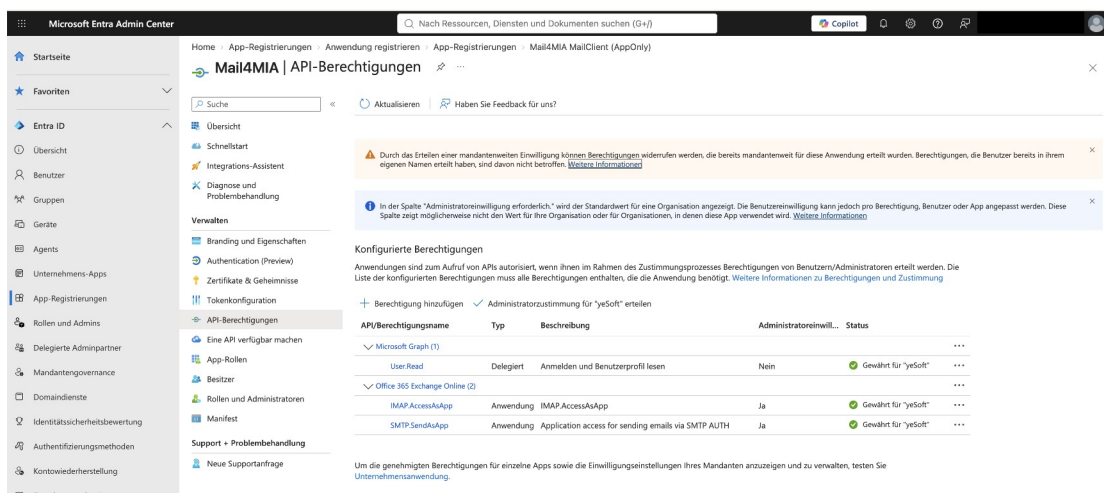


Schritt 2: Application Permissions vergeben

Wichtig: Es gibt zwei grundsätzlich verschiedene Berechtigungsarten. Für den interaktionsfreien Zugriff brauchen Sie die zweite Art – Anwendungsberechtigungen („Application“), nicht delegierte Berechtigungen („Delegated“).

1. In der App-Registrierung: API-Berechtigungen → Berechtigung hinzufügen.
2. Reiter „APIs, die meine Organisation verwendet“ → nach Office 365 Exchange Online suchen.
3. Anwendungsberechtigungen auswählen (nicht „Delegierte Berechtigungen“) und folgende hinzufügen:

Berechtigung	Zweck
IMAP.AccessAsApp	Abruf der E-Mails per IMAP
SMTP.SendAsApp	Versand der E-Mails per SMTP



Schritt 3: Admin Consent erteilen

Anwendungsberechtigungen wirken tenant-weit und können deshalb nur von einem Administrator bestätigt werden – einmalig, danach nie wieder.

1. Auf der Seite „API-Berechtigungen“ den Button „Administratorzustimmung für [Ihre Organisation] erteilen“ anklicken.
2. Bestätigen. Der Status aller Berechtigungen wechselt auf „Erteilt für ...“ (grüner Haken).

Hinweis: Ist der Button ausgegraut, fehlt Ihnen die Rolle *Globaler Administrator* oder *Anwendungsadministrator* – bitten Sie eine berechtigte Person darum.

Schritt 4: Client Secret erzeugen

1. Zertifikate & Geheimnisse → Client-Geheimnisse → Neues Client-Geheimnis. (Je nach Sprache Ihres Portals heißt dies auch „Geheimer Clientschlüssel“ – das ist derselbe Menteintrag, nur eine andere Übersetzung.)
2. Beschreibung vergeben, Ablauf: maximal 24 Monate wählbar („Nie ablaufend“ steht in Entra ID nicht mehr zur Verfügung).
3. Den angezeigten Wert (nicht die „Geheimnis-ID“) sofort kopieren – er wird danach nicht mehr angezeigt.

Ablaufdatum merken: 24 Monate ist das Maximum, kein Vorschlag. Tragen Sie das Ablaufdatum in Ihren eigenen Kalender ein und informieren rechtzeitig vorher den Mailclient-Betreiber, damit ein neues Secret hinterlegt werden kann, bevor der Versand/Abwurf ausfällt.

Microsoft Entra Admin Center

Home > App-Registrierungen > Mail4MIA MailClient (AppOnly)

Mail4MIA MailClient (AppOnly) | Zertifikate & Geheimnisse

Suche

Haben Sie Feedback für uns?

Übersicht

Schnellstart

Integrations-Assistent

Diagnose und Problembehandlung

Verwalten

Branding und Eigenschaften

Authentication (Preview)

Zertifikate & Geheimnisse

Tokenkonfiguration

API-Berechtigungen

Eine API verfügbar machen

App-Rollen

Besitzer

Rollen und Administratoren

Manifest

Support + Problembehandlung

Neue Supportanfrage

Haben Sie einen Moment, um uns Feedback zu geben? →

Anhand von Anmeldeinformationen können vertrauliche Anwendungen sich beim Authentifizierungsdienst identifizieren, wenn sie Token (über ein HTTPS-Schema) an einem adressierbaren Webspeicherort erhalten. Für eine höhere Sicherheitsstufe wird empfohlen, ein Zertifikat (anstelle eines Clientgeheimnisses) als Anmeldeinformation zu verwenden.

Anwendungsregistrierungszertifikate, Geheimnisse und Verbundanmeldeinformationen finden Sie auf den Registerkarten unten.

Zertifikate (0) **Geheime Clientschlüssel (2)** Verbundanmeldeinformationen (0)

Eine geheime Zeichenfolge, die von der Anwendung beim Anfordern eines Tokens als Identitätsnachweis verwendet wird. Wird auch als Anwendungskennwort bezeichnet.

+ Neuer geheimer Clientschlüssel

Beschreibung	Gültig bis	Wert	Geheime ID
MIAClientKey	20.9.2028	6e88Q--lbfCxwTzwunZg19c-7eWuAaIM...	0c6b051b-a914-4422-bced-7593647a1a...

Sofort kopieren!

Schritt 5: Service Principal in Exchange registrieren (NEU)

Voraussetzung: Exchange-Online-PowerShell mit dem Modul ExchangeOnlineManagement (siehe Anhang A für die Einrichtung auf dem Mac, falls noch nicht vorhanden).

1. Mit Exchange Online verbinden:

```
Connect-ExchangeOnline -UserPrincipalName ihre.admin-adresse@ihredomain.de
```

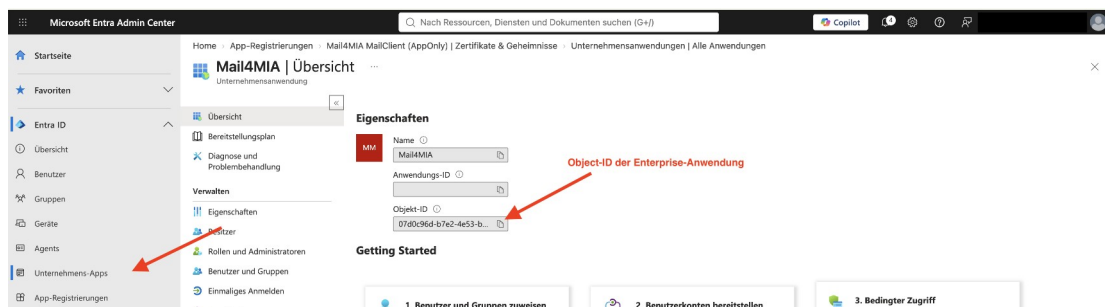
1. Prüfen, ob schon ein Service Principal existiert:

```
Get-ServicePrincipal | fl
```

Falls dort Ihre Anwendungs-ID nicht auftaucht (oder gar keine Ausgabe kommt), muss neu registriert werden.

Wichtig – die richtige Objekt-ID finden: Für die Registrierung wird die Objekt-ID der Enterprise-Anwendung benötigt – das ist eine andere ID als die Objekt-ID der App-Registrierung aus Schritt 1! Die Verwechslung ist laut Microsoft-Dokumentation der häufigste Fehler an dieser Stelle.

1. Entra Admin Center → Enterprise-Anwendungen (nicht „App-Registrierungen“) → nach Ihrer App suchen (Name oder Anwendungs-ID).
2. Übersichtsseite öffnen → Objekt-ID kopieren.



Service Principal registrieren:

```
New-ServicePrincipal -AppId "<Anwendungs-ID aus Schritt 1>" \  
-ObjectId "<Objekt-ID der Enterprise-Anwendung>" \  
-DisplayName "mailclient4mia App-Only"
```

Kontrolle:

```
Get-ServicePrincipal | fl
```

Hinweis: Merken Sie sich die zurückgegebene ObjectId/Identity des Service Principal (z. B. eine GUID) – sie wird im nächsten Schritt als Empfänger von Berechtigungen gebraucht.

Schritt 6: Postfach-Berechtigungen für den Service Principal setzen (NEU)

Für IMAP-Zugriff: FullAccess auf das Postfach

```
Add-MailboxPermission -Identity "postfach@ihredomain.de" \  
-User "mailclient4mia App-Only" \  
-AccessRights FullAccess
```

Für SMTP-Versand: SendAs-Recht

```
Add-RecipientPermission -Identity "postfach@ihredomain.de" \  
-Trustee "mailclient4mia App-Only" \  
-AccessRights SendAs
```

(„mailclient4mia App-Only“ ist hier der DisplayName des Service Principal aus Schritt 5 – alternativ funktioniert auch dessen ObjectId/Identity.)

Hinweis: Wie bei den meisten Berechtigungsänderungen in Exchange Online kann es bis zu 60 Minuten dauern, bis diese Rechte tatsächlich wirksam werden – auch wenn PowerShell die Änderung sofort bestätigt. Nicht sofort weitertesten und an weiteren Einstellungen drehen, sondern warten und dann erneut prüfen.

Schritt 7: Werte an den Mailclient-Betreiber übergeben

Übermitteln Sie die folgenden Werte auf einem sicheren Weg (z. B. verschlüsselte Mail, Passwort-Tresor) – niemals unverschlüsselt per Klartext-Mail oder Chat:

- Tenant-ID (Verzeichnis-ID) – aus der App-Übersichtsseite
- Client-ID (Anwendungs-ID) – aus der App-Übersichtsseite
- Client Secret (Wert aus Schritt 4) – nur einmalig sichtbar, jetzt weitergeben
- Ablaufdatum des Client Secrets (aus Schritt 4) – optional, aber empfohlen; kann im mailclient4mia-Kontoformular hinterlegt werden

Diese Werte werden dort im Konto-Datensatz des jeweiligen Postfachs hinterlegt und ersetzen die bisherige Geräte-Code-Anmeldung.

Schritt 8: Zugriff auf die richtigen Postfächer beschränken

Sicherheitskritischer Schritt – nicht überspringen Ohne diesen Schritt darf die App per Application Permission auf alle Postfächer im gesamten Tenant zugreifen, nicht nur auf die tatsächlich genutzten. Mit einer Application Access Policy schränken Sie den Zugriff gezielt auf die benötigten Postfächer ein.

Fall A: Die App bedient nur ein einziges Postfach

Dann wird keine Gruppe benötigt – die Mailbox-Adresse kann direkt als PolicyScopeGroupId angegeben werden:

```
New-ApplicationAccessPolicy -AppId "<Anwendungs-ID>" \  
-PolicyScopeGroupId "postfach@ihredomain.de" \  
-AccessRight RestrictAccess \  
-Description "mailclient4mia App-Only Zugriff"
```

Fall B: Die App bedient mehrere Postfächer

Legen Sie zuerst eine mail-aktivierte Sicherheitsgruppe an und nehmen Sie alle betroffenen Postfächer als Mitglieder auf, statt für jedes Postfach eine eigene Policy zu pflegen. Das hat den Vorteil, dass später neue Postfächer einfach der Gruppe hinzugefügt werden können, ohne die Policy selbst anzufassen:

```
New-DistributionGroup -Name "mailclient4mia-Postfaecher" -Type Security \  
-Members postfach1@ihredomain.de,postfach2@ihredomain.de  
  
New-ApplicationAccessPolicy -AppId "<Anwendungs-ID>" \  
-PolicyScopeGroupId "mailclient4mia-Postfaecher" \  
-AccessRight RestrictAccess \  
-Description "mailclient4mia App-Only Zugriff (mehrere Postfaecher)"
```

Hinweis: Der Gruppenname muss eindeutig sein – meldet PowerShell „Es sind mehrere Empfänger vorhanden, die der Identität entsprechen“, existiert bereits ein Objekt mit diesem Namen (z. B. aus einem früheren Versuch); prüfen Sie mit `Get-Recipient -Identity „<Name>“`, was da schon existiert, oder wählen Sie einen anderen Namen.

Verifikation der Policy

```
Test-ApplicationAccessPolicy -AppId "<Anwendungs-ID>" -Identity  
"postfach@ihredomain.de"
```

Das Ergebnis muss AccessCheckResult: Granted (bzw. „Gewährt“) zeigen.

Hinweis: Gibt `Get-ApplicationAccessPolicy` einen Fehler wie „Das Objekt ... wurde nicht gefunden“ zurück, ist das kein Problem, sondern eine bekannte Eigenheit von Exchange Online: Diese Meldung erscheint, wenn noch gar keine Application Access Policy existiert – dann ist der Zugriff standardmäßig für alle Postfächer offen.

Schritt 9: Authentifiziertes SMTP tenant-weit aktivieren

Microsoft deaktiviert authentifiziertes SMTP-Client-Submission (AUTH auf Port 587) standardmäßig tenant-weit. Ohne diesen Schritt meldet der SMTP-Test den Fehler „SmtpClientAuthentication is disabled for the Tenant.“

Prüfen:

```
Get-TransportConfig | fl SmtpClientAuthenticationDisabled
```

Falls True, aktivieren:

```
Set-TransportConfig -SmtpClientAuthenticationDisabled $false
```

Zusätzlich gibt es eine mailbox-spezifische Überschreibung, die den Tenant-Wert überstimmen kann – auch die prüfen:

```
Get-CASMailbox -Identity "postfach@ihredomain.de" | fl  
SmtpClientAuthenticationDisabled
```

Falls dort True (oder ein blockierender Wert), ebenfalls deaktivieren:

```
Set-CASMailbox -Identity "postfach@ihredomain.de" -SmtpClientAuthenticationDisabled  
$false
```

Hinweis: Diese Änderung braucht laut Microsoft bis zu 60 Minuten, um wirksam zu werden – der PowerShell-Befehl bestätigt die Änderung sofort, das ist aber nicht dasselbe wie die tatsächliche Durchsetzung im Exchange-Online-Frontend.

Schritt 10: Protokoll-Freischaltung und Lizenz prüfen

IMAP/POP für das Postfach aktiviert?

```
Get-CASMailbox -Identity "postfach@ihredomain.de" | fl ImapEnabled, PopEnabled
```

Falls False:

```
Set-CASMailbox -Identity "postfach@ihredomain.de" -ImapEnabled $true
```

OAuth für IMAP/POP/SMTP auf Organisationsebene aktiviert?

```
Get-OrganizationConfig | fl OAuth2ClientProfileEnabled
```

Sollte True sein. Falls False:

```
Set-OrganizationConfig -OAuth2ClientProfileEnabled $true
```

Lizenz

Das Postfach benötigt eine aktive Lizenz mit Exchange Online – mindestens Exchange Online Plan 1 (im Admin Center unter Benutzer → Aktive Benutzer → Lizenzen und Apps prüfbar). Ohne Lizenz kann selbst ein korrekt als „ImapEnabled: True“ markiertes Postfach den Protokollzugriff verweigern.

Trial-/Testtenants: Ganz neu angelegte Tenants (insbesondere kostenlose Test-Abos) unterliegen erfahrungsgemäß zusätzlichen, von Microsoft nicht dokumentierten Anti-Missbrauchs-Einschränkungen in den ersten Tagen nach Anlage – selbst wenn sämtliche Konfigurationspunkte dieser Anleitung nachweislich korrekt gesetzt sind, kann die Authentifizierung dort tagelang fehlschlagen.

Anhang: Fehlerbehebung – Fehlermeldung → Ursache → Lösung

Diese Tabelle fasst die während der Ersteinrichtung tatsächlich aufgetretenen Fehlerbilder zusammen, in der Reihenfolge, in der sie typischerweise nacheinander auftreten, wenn Schritte fehlen.

Fehlermeldung	Wahrscheinliche Ursache → Lösung
„connection setup failed“ (mailclient4mia-Protokoll)	Generische Sammelmeldung von webklex/php-imap – die tatsächliche Ursache steht nur im PHP-Error-Log. Dort nach der Zeile „got failure response: ...“ suchen, die die echte Serverantwort enthält.
„NO AUTHENTICATE failed.“	Häufigste Ursache: Service Principal fehlt (Schritt 5). Weitere mögliche Ursachen: Application Permission/Admin Consent unvollständig (Schritt 2–3), SmtplibClientAuthenticationDisabled=True (Schritt 9, betrifft auch IMAP-nahe Symptome indirekt), oder bei sehr jungen Tenants eine temporäre Anti-Missbrauchs-Sperre (Schritt 10).
„SmtplibClientAuthentication is disabled for the Tenant.“	Authentifiziertes SMTP ist tenant-weit deaktiviert (Microsoft-Standard). Set-TransportConfig -SmtplibClientAuthenticationDisabled \$false setzen (Schritt 9), bis zu 60 Minuten Wartezeit einplanen.
„NO User is authenticated but not connected.“	Authentifizierung erfolgreich, aber der Service Principal hat kein Zugriffsrecht auf das Postfach selbst. Add-MailboxPermission ... -AccessRights FullAccess setzen (Schritt 6).
SMTP-Test besteht, aber der tatsächliche Versand schlägt fehl	SendAs-Recht fehlt. Add-RecipientPermission ... -AccessRights SendAs setzen (Schritt 6).
Alle Punkte dieser Anleitung nachweislich korrekt, Fehler besteht dennoch weiter	Propagationsverzögerung (bis zu 60 Minuten pro Änderung sind normal) oder, bei sehr jungen bzw. Trial-Tenants, eine mehrtägige Anti-Missbrauchs-Sperre von Microsoft. Keine weiteren Änderungen vornehmen, sondern abwarten und dann unverändert erneut testen.

Praktischer Tipp zur Eingrenzung: Ein manueller Test per openssl s_client (ganz ohne mailclient4mia) beweist zweifelsfrei, ob ein Fehler client- oder serverseitig liegt:

```
TOKEN="access_token_aus_temailauth"
USER="postfach@ihredomain.de"
AUTHSTR=$(printf 'user=%s\001auth=Bearer %s\001\001' "$USER" "$TOKEN" | base64 | tr -d '\n')
printf 'a1 AUTHENTICATE XOAUTH2 %s\r\n' "$AUTHSTR" > /tmp/imapcmd.txt

( cat /tmp/imapcmd.txt; sleep 5 ) | openssl s_client -connect outlook.office365.com:993 -crlf -quiet
```

Hinweis: Der Token wird dabei nie irgendwohin übertragen außer an den Microsoft-Server selbst – alles läuft lokal im Terminal.