

Klar, ich liste dir hier die Maildomains und Anbieter auf, die **wirklich zuverlässig mit Python-Bibliotheken wie `imaplib`, `smtplib`** usw. funktionieren. Dabei gehe ich davon aus, dass du gängige Standardprotokolle nutzt (IMAP/SMTP) mit den üblichen Ports (IMAP SSL 993, SMTP STARTTLS 587).

Maildomains/Provider, die zuverlässig mit IMAP & SMTP in Python funktionieren

1. Google Gmail

- IMAP-Server: `imap.gmail.com` (Port 993 SSL)
 - SMTP-Server: `smtp.gmail.com` (Port 587 STARTTLS)
 - Besonderheiten:
 - OAuth2 wird empfohlen, Passwort-Login oft blockiert (bzw. App-Passwort notwendig).
 - Zwei-Faktor-Authentifizierung (2FA) kann aktiviert sein.
 - Domains: `@gmail.com`, G Suite (Google Workspace) Domains.
-

2. Microsoft Outlook / Office365 / Hotmail

- IMAP-Server: `outlook.office365.com` oder `imap-mail.outlook.com` (Port 993 SSL)
 - SMTP-Server: `smtp.office365.com` oder `smtp-mail.outlook.com` (Port 587 STARTTLS)
 - Domains: `@outlook.com`, `@hotmail.com`, `@live.com`, eigene Office365-Domains
 - Besonderheiten:
 - OAuth2 wird unterstützt, meist nötig.
 - SMTP Auth muss aktiviert sein.
-

3. Yahoo Mail

- IMAP-Server: `imap.mail.yahoo.com` (Port 993 SSL)
- SMTP-Server: `smtp.mail.yahoo.com` (Port 587 STARTTLS)
- Domains: `@yahoo.com`, `@ymail.com`, `@rocketmail.com`

- Besonderheiten:
 - App-Passwörter für externe Apps notwendig, wenn 2FA aktiv.
-

4. GMX

- IMAP-Server: `imap.gmx.net` (Port 993 SSL)
 - SMTP-Server: `smtp.gmx.net` (Port 587 STARTTLS)
 - Domains: `@gmx.de`, `@gmx.net`
 - Besonderheiten:
 - Relativ unkompliziert, klassische Passwortauthentifizierung möglich.
-

5. Web.de

- IMAP-Server: `imap.web.de` (Port 993 SSL)
 - SMTP-Server: `smtp.web.de` (Port 587 STARTTLS)
 - Domains: `@web.de`
 - Besonderheiten:
 - Passwortauthentifizierung, 2FA möglich.
-

6. ProtonMail (via Bridge)

- ProtonMail selbst bietet keinen direkten IMAP/SMTP Zugriff.
 - **Man braucht ProtonMail Bridge**, die lokal als Mailserver agiert.
 - Domains: `@protonmail.com`
 - Besonderheiten:
 - Nicht direkt IMAP/SMTP, Bridge läuft lokal und leitet weiter.
-

7. Zoho Mail

- IMAP-Server: `imap.zoho.com` (Port 993 SSL)
 - SMTP-Server: `smtp.zoho.com` (Port 587 STARTTLS)
 - Domains: `@zoho.com`, eigene Domains via Zoho
 - Besonderheiten:
 - Passwortauthentifizierung oder OAuth2 möglich.
-

8. FastMail

- IMAP-Server: `imap.fastmail.com` (Port 993 SSL)
 - SMTP-Server: `smtp.fastmail.com` (Port 587 STARTTLS)
 - Domains: eigene oder `@fastmail.com`
 - Besonderheiten:
 - Stabile und gut dokumentierte Server.
-

9. Yahoo Business / Verizon

- Funktioniert ähnlich wie Yahoo Mail, aber evtl. eigene Server.
-

10. Mail.com

- IMAP-Server: `imap.mail.com` (Port 993 SSL)
 - SMTP-Server: `smtp.mail.com` (Port 587 STARTTLS)
 - Domains: `@mail.com`
 - Besonderheiten:
 - Klassische Authentifizierung.
-

Was funktioniert eher nicht oder nur sehr kompliziert?

- Anbieter, die nur Webmail (z.B. Outlook.com Web) oder eigene Protokolle verwenden.
 - E-Mail-Provider, die OAuth2 erzwingen, ohne einfache App-Passwörter (z.B. Google, Microsoft – da OAuth2 aber unterstützt wird, ist es technisch möglich, aber komplex).
 - Anbieter mit CAPTCHAs oder Anti-Bot-Mechanismen (z.B. einige kleine Provider).
 - E-Mail-Adressen, die auf sehr exotischen oder privaten Servern laufen (können funktionieren, erfordern aber manuelle Konfiguration).
-

Zusammenfassung / Empfehlung

Provider	IMAP Server	SMTP Server	Bemerkung
Gmail	<code>imap.gmail.com</code>	<code>smtp.gmail.com</code>	OAuth2, App-Passwort nötig
Outlook / Office365	<code>outlook.office365.com</code>	<code>smtp.office365.com</code>	OAuth2 empfohlen

Provider	IMAP Server	SMTP Server	Bemerkung
Yahoo	imap.mail.yahoo.com	smtp.mail.yahoo.com	App-Passwort bei 2FA
GMX	imap.gmx.net	smtp.gmx.net	Einfach, klassisch
Web.de	imap.web.de	smtp.web.de	Einfach, klassisch
Zoho Mail	imap.zoho.com	smtp.zoho.com	Einfach
FastMail	imap.fastmail.com	smtp.fastmail.com	Sehr zuverlässig
Mail.com	imap.mail.com	smtp.mail.com	Einfach

Falls du noch Domains/Provider für spezielle Länder brauchst, sag Bescheid, ich ergänze die Liste gern.

Noch ein Tipp für dein Programm:

- Implementiere eine Datenbank oder Config mit vordefinierten Serverdaten für gängige Domains.
- Erkenne anhand der eingegebenen E-Mail-Adresse die Domain und setze automatisch die Server/Ports (z.B. bei @gmail.com etc.).
- Biete die Möglichkeit, diese automatisch gesetzten Werte zu überschreiben.

Wenn du willst, kann ich dir auch helfen, so eine Mapping-Tabelle in Python aufzubauen!