

## SOGGo SSO API Benutzerhandbuch

Dieses Dokument beschreibt die Nutzung der bereitgestellten SOGo SSO API-Endpunkte `sogo-auth.php` und `sogo-tokengenerate.php`. Diese API ermöglicht die Authentifizierung und Tokenverwaltung für Benutzer im Zusammenhang mit dem SOGo-Dienst.

---

### 1. Endpunkte

#### 1.1 sogo-auth.php

- **URL:** `/sogossologin/sogo-auth.php`
- **Methode:** GET
- **Request-Typ:** GET-Parameter
- **Parameter:**
  - `email` (erforderlich): Die E-Mail-Adresse des Benutzers.
  - `token` (erforderlich): Der für den Benutzer generierte Token.
- **Beschreibung:**
  - Der Nutzer wird von der eigenen Anwendung auf diese Seite geleitet, um den Token zu überprüfen.
  - Dieser Endpunkt überprüft, ob der bereitgestellte Token für den Benutzer in der Datenbank existiert.
  - Bei erfolgreicher Authentifizierung wird der Benutzer zu seiner SOGo-Seite weitergeleitet.
  - Fehlgeschlagene Authentifizierungen geben einen HTTP-Statuscode 401 zurück.

#### Beispiel-Anfrage:

```
GET https://mail.example.com/sogossologin/sogo-  
auth.php?email=user@example.com&token=abc123 HTTP/1.1  
Host: example.com
```

#### Mögliche Antworten:

- **200 OK:** Erfolgreiche Authentifizierung, Weiterleitung zur SOGo-Oberfläche.
  - **401 Unauthorized:** Ungültiger Token oder fehlgeschlagene Authentifizierung.
- 

#### 1.2 sogo-tokengenerate.php

- **URL:** `/sogossologin/sogo-tokengenerate.php`
- **Methode:** POST
- **Header:**
  - `Content-Type: application/json`
- **Body-Parameter:**
  - `username` (erforderlich): Der Benutzername, für den der Token generiert werden soll.

- apikey (erforderlich): Der API-Schlüssel (Aus dem Mailcow-Dashboard) zur Authentifizierung der Anfrage.
- **Beschreibung:**
  - Erstellt einen neuen Token für den Benutzer und speichert ihn in der Datenbank.
  - Gibt den generierten Token als JSON-Antwort zurück.

#### Beispiel-Anfrage (PHP):

```
public function generateSOGoLoginToken($email) {
    $url = "https://mail.example.com/sogossologin/sogo-tokengenerate.php";

    $curl = curl_init($url);
    curl_setopt($curl, CURLOPT_POST, true);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);

    $headers = ["Content-Type: application/json"];
    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);

    $data = json_encode([
        "username" => $email,
        "apikey" => $this->apikey
    ]);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);

    // Debug-Optionen nur in Entwicklungsumgebungen verwenden
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);

    $resp = curl_exec($curl);
    curl_close($curl);

    $responseData = json_decode($resp, true);
    if ($responseData['success']) {
        return $responseData['token'];
    }

    return null;
}
```

#### Beispiel-Antwort:

```
{
  "success": true,
  "username": "user@example.com",
  "token": "f1d2d2f924e986ac86fdf7b36c94bcd32beec15"
}
```

#### Mögliche Fehler:

- **401 Unauthorized:** Ungültiger API-Key.
- **500 Internal Server Error:** Datenbankfehler.

---

## 2. Anforderungen

- PHP-Unterstützung auf dem Server.
- Zugriff auf die SOGo-Datenbank.
- Korrekt konfigurierte Datenbankverbindung in `prerequisites.inc.php`.

## 3. Sicherheitshinweise

- API-Schlüssel sollten sicher aufbewahrt und nicht im Klartext übermittelt werden.
- Stellen Sie sicher, dass die Datenbankverbindung sicher und nur für autorisierte Systeme erreichbar ist.
- HTTPS sollte für alle Anfragen verwendet werden, um die Sicherheit der Kommunikation zu gewährleisten.

---

## 4. Kontakt

Bei Problemen oder Fragen können Sie mich gerne unter [me@ysenayit.de](mailto:me@ysenayit.de) kontaktieren.