

# ☐☐ API-Token in Proxmox erstellen

title:: "ChatGPT" source:: "https://chatgpt.com/c/68b16db1-0144-8322-86c2-76a0c43f9e30" author:: [[ChatGPT]] published:: created:: 2025-08-29 description:: "ChatGPT hilft dir, Antworten zu finden, dich inspirieren zu lassen und produktiver zu sein. Du kannst es ganz einfach kostenlos ausprobieren. Stelle irgendeine Frage und ChatGPT kann dir beim Schreiben, Lernen, Ideensammeln und mehr helfen." tags:: clippings

## 1. Anmelden

- Melde dich auf der **Proxmox WebGUI** an (`https://<dein-host>:8006`).

## 2. Benutzer auswählen

- Links im Baum: **Rechenzentrum** → **Berechtigungen** → **Benutzer**.
- Wähle den Benutzer, unter dem du arbeiten willst, z. B. `root@pam`.
- ☐☐ Für Automatisierungen empfiehlt es sich, einen separaten Benutzer anzulegen, z. B. `automation@pve`.

## 3. Token anlegen

- Klick auf den Benutzer → Tab **API Token** → **Hinzufügen**.
- Gib ein:
- **Token-ID** (frei wählbarer Name, z. B. `md-gen`).
- Häkchen bei **Unverfallbar** setzen, wenn das Token nicht automatisch ablaufen soll.
- Optional: Ablaufdatum, wenn du es zeitlich begrenzen willst.
- Bestätigen mit **Hinzufügen**.

## 4. Rechte vergeben

- Das Token selbst hat keine Rechte, die musst du separat vergeben.
- Gehe zu: **Rechenzentrum** → **Berechtigungen** → **Berechtigungen** → **Hinzufügen** → **API Token Berechtigung**.
- Wähle:
- **Pfad:** / (oder gezielt /vms wenn du nur VM-Zugriff brauchst).
- **API Token:** deinen Benutzer + das neu erstellte Token.
- **Rolle:** mindestens PVEVM.Audit und Sys.Audit (damit kann man Configs und Status lesen). Falls du später auch Snapshots oder Backups auslesen willst: VM.Backup / Datastore.Audit ergänzen.
- Mit **Hinzufügen** speichern.

## 5. Token-Secret notieren

- Nach dem Erstellen wird dir **Token-ID** und **Secret** angezeigt.
- ⚠ Das Secret siehst du **nur einmal!** Gleich kopieren und sicher ablegen.

## 📄 Ergebnis

- Dein API-Token besteht aus:
- Benutzer: root@pam oder z. B. automation@pve
- Token-ID: md-gen
- Token-Secret: ein langer Schlüssel (UUID-ähnlich)

Im Script/Environment trägst du es so ein:

```
export PMX_HOST="proxmox.dein.tld:8006"
export PMX_USER="root@pam"
export PMX_TOKEN_NAME="md-gen"
export PMX_TOKEN_VALUE="xxxxxxxx-xxxx-xxxx-xxxx"
export VERIFY_SSL="true"
```

📄 Soll ich dir auch gleich ein Beispiel geben, wie du einen **eigenen Service-User** anlegst (anstatt root@pam zu benutzen)? Das ist meist die sicherere Variante.

Du:

ja bitte

ChatGPT:

Perfekt `[]` - einen **eigenen Service-User** zu verwenden ist deutlich sicherer als `root@pam`.

Hier die Schritte:

---

# `[]` Eigenen Proxmox Service-User mit API-Token erstellen

## 1. Benutzergruppe für Automation anlegen

1. In der WebGUI: **Rechenzentrum** → **Berechtigungen** → **Gruppen**.
2. **Gruppe hinzufügen** - z. B. Name `automation`.

## 2. Benutzer anlegen

1. Gehe zu **Rechenzentrum** → **Berechtigungen** → **Benutzer**.
2. **Hinzufügen**:

- **Benutzername:** `automation`
- **Realm:** `pve` (damit ist es ein interner Proxmox-User, kein PAM/LDAP).
- **Kommentar:** „Automation Account für API-Zugriff“
- Passwort kannst du setzen, ist aber für Token-Nutzung nicht zwingend.

3. Benutzer zur Gruppe `automation` hinzufügen.

## 3. Rechte für die Gruppe vergeben

1. Gehe zu **Rechenzentrum** → **Berechtigungen** → **Berechtigungen**.
2. **Hinzufügen** → **Gruppe Berechtigung**:

- **Pfad:** `/` (für kompletten Zugriff lesen - alternativ `/vms` oder `/storage` enger einschränken).
- **Gruppe:** `automation`.

- **Rolle:** mindestens:
- `PVEVM.Audit` (VM-Infos lesen)
- `Sys.Audit` (Cluster/Node Infos lesen)
- Optional: `Datastore.Audit` (wenn du Storage/Backup-Infos brauchst)

☐☐ Damit kann der Account **lesen**, keine VM starten/stoppen.

## 4. API-Token für den User erzeugen

1. Wieder zu **Rechenzentrum** → **Berechtigungen** → **Benutzer**.
2. Wähle den User `automation@pve`.
3. Tab **API Token** → **Hinzufügen**:

- **Token-ID:** z. B. `md-gen`.
- Häkchen bei **Unverfallbar** setzen (außer du willst ein Ablaufdatum).

4. Nach dem Erstellen bekommst du:

- **Token-ID:** `automation@pve!md-gen`
- **Token-Secret:** einmalig sichtbar → sofort kopieren!

## 5. Token ins Script eintragen

In deiner Shell/Umgebung:

```
export PMX_HOST="proxmox.dein.tld:8006"
export PMX_USER="automation@pve"
export PMX_TOKEN_NAME="md-gen"
export PMX_TOKEN_VALUE="xxxxxxxx-xxxx-xxxx-xxxx"
export VERIFY_SSL="true"
```

---

## ☐☐ Vorteil

- Kein Root-Zugriff nötig

- Rechte sauber begrenzt (nur Lesen)
- Token kann jederzeit widerrufen oder neu erstellt werden

---

☐ Soll ich dir ein **Rollenprofil** bauen (z. B. `VM.DokuLesen`), das nur die **minimal nötigen Rechte** fürs Dokumentations-Skript hat? Dann kannst du es an den Automation-User binden, ohne zu viel freizugeben.

Du:

Ja bitte

ChatGPT:

Super — hier ist ein **minimal sicheres Rollenprofil** für dein Doku-Skript. Es erlaubt reines Lesen von Cluster/VM/Storage-Infos und (optional) QEMU-Guest-Agent-Abfragen für IPs.

## ☐ Rolle „VM.DokuLesen“

### Enthaltene Privilegien

- **Sys.Audit** → Cluster/Nodes lesen
- **PVEVM.Audit** → VM-Konfig & Status lesen (inkl. Snapshots)
- **Datastore.Audit** → Storage/Volumes lesen (nur lesend)
- **VM.Monitor** (*optional, empfohlen*) → nötig für **Guest-Agent** API (`network-get-interfaces`), um IPs auszulesen

“ Wenn du **keine** IPs via Guest-Agent brauchst, lass **VM.Monitor** weg.

---

## Variante A: Klickpfad (GUI)

### 1. Rechenzentrum → Berechtigungen → Rollen → Hinzufügen

- **Name:** `VM.DokuLesen`
- **Rechte anhaken:** `Sys.Audit`, `PVEVM.Audit`, `Datastore.Audit`, optional `VM.Monitor`
- **Speichern**

## 2. Rechenzentrum → Berechtigungen → Berechtigungen → Hinzufügen → Gruppe Berechtigung

- **Pfad:** / (oder enger, z. B. /vms und /storage)
- **Gruppe:** automation
- **Rolle:** VM.DokuLesen
- **Hinzufügen**

(Du hast die Gruppe/den User automation@pve ja bereits erstellt.)

# Variante B: CLI (schnell & reproduzierbar)

```
# Rolle anlegen (mit Guest-Agent-Recht)
pveum roleadd VM.DokuLesen -privs "Sys.Audit,PVEVM.Audit,Datastore.Audit,VM.Monitor"

# Falls ohne Guest-Agent:
# pveum roleadd VM.DokuLesen -privs "Sys.Audit,PVEVM.Audit,Datastore.Audit"

# Gruppe (falls noch nicht vorhanden)
pveum groupadd automation -comment "Automation Gruppe"

# User (falls noch nicht vorhanden)
# pveum useradd automation@pve -password <GEHEIM> -comment "Automation User"

# Rolle der Gruppe auf / zuweisen (Read-only, clusterweit)
pveum aclmod / -group automation -role VM.DokuLesen
```

# Sicherheits-Hinweise

- **Least Privilege:** Weisen, wenn möglich, nur auf die benötigten Teilpfade zu (z. B. nur bestimmter Node/Pool/Storage).
- **Tokens statt Passwort:** Für das Skript ausschließlich **API-Token** von automation@pve verwenden.
- **Rotation:** Token mit Ablaufdatum erstellen oder regelmäßig neu ausstellen.

- **Audit:** Änderungen an Rechten protokollieren (z. B. Änderungs-Ticket).
- 

Wenn du willst, gebe ich dir noch eine „**Härtungs-Checkliste**“ (Self-Signed TLS, Token-Ablauf, engere ACL-Pfadwahl pro Pool/Node, Logging), die du einfach abarbeiten kannst.

---

Revision #1

Created 29 August 2025 10:55:03 by Ahorn

Updated 29 August 2025 10:58:59 by Ahorn