

SECRETS MANAGER > ENTWICKLERTOOLS

Secrets Manager CLI herunter

Ansicht im Hilfezentrum:

<https://bitwarden.com/help/secrets-manager-cli/>

Secrets Manager CLI herunter

Die Secrets Manager Kommandozeilen-Schnittstelle (CLI) ist ein leistungsstarkes Werkzeug zum Abrufen und Einspritzen Ihrer Geheimnisse. Der Secrets Manager CLI kann verwendet werden, um Ihren Tresor mit **erstellen**, **löschen**, **bearbeiten** und **auflisten** Ihrer Geheimnisse und Projekte zu organisieren.

Der Secrets Manager CLI ist selbst dokumentiert. Von der Befehlszeile aus erfahren Sie mehr über die verfügbaren Befehle mit:

Bash

```
bws --help, -h
```

Herunterladen und installieren

Die CLI kann plattformübergreifend auf Windows, macOS und Linux-Distributionen verwendet werden. Um den Secrets Manager CLI herunterzuladen und zu installieren:

Laden Sie die Secrets Manager CLI von <https://github.com/bitwarden/sdk/releases> herunter.

Authentifizierung

Der Secrets Manager CLI kann mit einem für ein bestimmtes **Service-Konto** generierten **Zugriffs-Token** angemeldet werden. Das bedeutet, dass **nur Geheimnisse und Projekte, auf die das Dienstkonto Zugriff hat**, mit der CLI interagiert werden dürfen. Es gibt einige Möglichkeiten, wie Sie eine CLI-Sitzung authentifizieren können:

⇒Umgebungsvariable

Sie können eine CLI-Sitzung authentifizieren, indem Sie eine Umgebungsvariable **BWS_ACCESS_TOKEN** mit dem Wert Ihres Zugriffstokens speichern, zum Beispiel:

Bash

```
export BWS_ACCESS_TOKEN=0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprlPOYutt0:B3h5D+YgLvFiQhWkIq6Bow==
```

⇒In der Reihe

Sie können einzelne CLI-Anfragen mit dem **-t**, **--access-token** Flag bei jedem einzelnen Befehl authentifizieren, zum Beispiel:

Bash

```
bws secret list --access-token 0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprlPOYutt0:B3h5D+YgLvFiQhWkIq6Bow==
```

⚠ Warning

Wenn Ihr Arbeitsablauf viele separate Sitzungen verwendet (wobei jede Verwendung eines Zugriffstokens zur Authentifizierung eine "Sitzung" darstellt), um Anfragen von derselben IP-Adresse in kurzer Zeit zu stellen, können Sie auf Geschwindigkeitsbegrenzungen stoßen.

Befehle

Befehle werden verwendet, um mit dem Secrets Manager CLI zu interagieren. Geheimnisse und Projekte können je nach den Berechtigungen, die Ihrem spezifischen Zugriffs-Token gegeben wurden, gelesen oder geschrieben werden. Für zusätzliche Details zu den verfügbaren Befehlen für `Geheimnis` und `Projekt`, verwenden Sie:

- `bws Geheimnis --Hilfe`
- `bws Projekt --Hilfe`

📘 Note

Ab der Secrets Manager Version 0.3.0 wurde die CLI-Syntax geändert. Der Befehl zum Auflisten von Geheimnissen hat sich beispielsweise von `bws list secrets` zu `bws secret list` geändert.

Die alte Syntax wird vorübergehend in der Secrets Manager CLI unterstützt bleiben. Wenn Sie nicht sicher sind, welche Version des Secrets Manager CLI Sie verwenden, geben Sie `bws --version` ein.

Geheimnis

Der `Geheim` Befehl wird verwendet, um auf [Geheimnisse](#) zuzugreifen, sie zu manipulieren und zu erstellen. Wie bei allen Befehlen können Geheimnisse und [Projekte](#), die außerhalb des Zugriffsbereichs Ihres Zugriffstokens liegen, nicht gelesen oder beschrieben werden.

geheimes erstellen

Verwenden Sie `bws secret create`, um ein neues Geheimnis zu erstellen. Dieser Befehl erfordert einen `SCHLÜSSEL`, einen `WERT` und eine `PROJEKT_ID`:

Bash

```
bws secret create <KEY> <VALUE> <PROJECT_ID>
```

Optional können Sie eine Notiz hinzufügen, indem Sie die Option `--note` verwenden. Zum Beispiel:

Bash

```
bws secret create SES_KEY 0.982492bc-7f37-4475-9e60 f588b2f2-4780-4a78-be2a-b02d014d622f --note "API Key for AWS SES"
```

Dieser Befehl gibt standardmäßig ein JSON-Objekt zurück und speichert das Geheimnis im Secrets Manager. Sie können das Ausgabeformat mit der `--output` Flagge ändern ([mehr erfahren](#)).

Bash

```
{
  "object": "secret",
  "id": "be8e0ad8-d545-4017-a55a-b02f014d4158",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "projectId": "e325ea69-a3ab-4dff-836f-b02e013fe530",
  "key": "SES_KEY",
  "value": "0.982492bc-7f37-4475-9e60",
  "note": "API Key for AWS SES",
  "creationDate": "2023-06-28T20:13:20.643567Z",
  "revisionDate": "2023-06-28T20:13:20.643567Z"
}
```

geheimes Löschen

Verwenden Sie `bws secret löschen`, um ein oder mehrere durch die `SECRET_IDS` bezeichnete Geheimnisse zu löschen.

Bash

```
bws secret delete <SECRET_IDS>
```

Um ein einzelnes Geheimnis mit der `id` `be8e0ad8-d545-4017-a55a-b02f014d4158` zu löschen:

Bash

```
bws secret delete be8e0ad8-d545-4017-a55a-b02f014d4158
```

Für mehrere Geheimnisse, bei denen die `IDs` `382580ab-1368-4e85-bfa3-b02e01400c9f` und `47201c5c-5653-4e14-9007-b02f015b2d82` sind:

Bash

```
bws secret delete 382580ab-1368-4e85-bfa3-b02e01400c9f 47201c5c-5653-4e14-9007-b02f015b2d82
```

Ausgabe:

Bash

```
1 secret deleted successfully.
```

geheime Bearbeitung

Um ein Geheimnis zu bearbeiten, wird die folgende Struktur Änderungen am ausgewählten Wert vornehmen. Von der CLI können diese Befehle den geheimen **SCHLÜSSEL**, **WERT**, **NOTIZ** oder **PROJEKT_ID** bearbeiten.

Bash

```
bws secret edit <SECRET_ID> --key <KEY> --value <VALUE> --note <NOTE> --project-id <PROJECT_ID>
```

Zum Beispiel, wenn Sie eine Notiz zu einem bestehenden Geheimnis hinzufügen möchten:

Bash

```
bws secret edit be8e0ad8-d545-4017-a55a-b02f014d4158 --note "I am adding a note"
```

Note

Setzen Sie Anführungszeichen um die Zeichenkette, wenn Sie eine **NOTIZ** bearbeiten, die Leerzeichen enthält.

Mehrere Felder bearbeiten, wo **SES_KEY2** der neue **Schlüssel** ist und **0.1982492bc-7f37-4475-9e60** der neue **Wert** ist:

Bash

```
bws secret edit be8e0ad8-d545-4017-a55a-b02f014d4158 --key SES_KEY2 --value 0.1982492bc-7f37-4475-9e60
```

Ausgabe:

Bash

```
{
  "object": "secret",
  "id": "be8e0ad8-d545-4017-a55a-b02f014d4158",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "projectId": "e325ea69-a3ab-4dff-836f-b02e013fe530",
  "key": "SES_KEY2",
  "value": "0.1982492bc-7f37-4475-9e60",
  "note": "I am adding a note",
  "creationDate": "2023-06-28T20:13:20.643567Z",
  "revisionDate": "2023-06-28T20:45:37.46232Z"
}
```

geheimes bekommen

Verwenden Sie `bws secret get`, um ein bestimmtes Geheimnis abzurufen:

Bash

```
bws secret get <SECRET_ID>
```

Standardmäßig ruft dieser Befehl das geheime Objekt mit der `SECRET_ID` ab.

Bash

```
bws secret get be8e0ad8-d545-4017-a55a-b02f014d4158
```

Standardmäßig gibt `get` Objekte als JSON-Array zurück, wie im folgenden Beispiel gezeigt. Sie können das Ausgabeformat ändern, indem Sie die `--output` Flagge verwenden ([mehr erfahren](#)).

Bash

```
{
  "object": "secret",
  "id": "be8e0ad8-d545-4017-a55a-b02f014d4158",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "projectId": "e325ea69-a3ab-4dff-836f-b02e013fe530",
  "key": "SES_KEY",
  "value": "0.982492bc-7f37-4475-9e60",
  "note": "",
  "creationDate": "2023-06-28T20:13:20.643567Z",
  "revisionDate": "2023-06-28T20:13:20.643567Z"
}
```

geheime Liste

Um die Geheimnisse aufzulisten, auf die das Service-Konto zugreifen kann, verwenden Sie den folgenden Befehl:

Bash

```
bws secret list
```

Sie können auch nur die Geheimnisse in einem bestimmten Projekt auflisten, indem Sie den folgenden Befehl verwenden, wobei `e325ea69-a3ab-4dff-836f-b02e013fe530` einen Projektidentifikator darstellt:

Bash

```
bws secret list e325ea69-a3ab-4dff-836f-b02e013fe530
```

Standardmäßig gibt `Liste` Objekte als JSON-Array zurück, wie im folgenden Beispiel. Sie können das Ausgabeformat ändern, indem Sie die `--output` Flagge verwenden ([mehr erfahren](#)).

Bash

```
[
  {
    "object": "secret",
    "id": "382580ab-1368-4e85-bfa3-b02e01400c9f",
    "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
    "projectId": "e325ea69-a3ab-4dff-836f-b02e013fe530",
    "key": "Repository 1",
    "value": "1234567ertthrytkuy",
    "note": "Main Repo",
    "creationDate": "2023-06-27T19:25:15.822004Z",
    "revisionDate": "2023-06-27T19:25:15.822004Z"
  },
  {
    "object": "secret",
    "id": "be8e0ad8-d545-4017-a55a-b02f014d4158",
    "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
    "projectId": "e325ea69-a3ab-4dff-836f-b02e013fe530",
    "key": "SES_KEY",
    "value": "0.982492bc-7f37-4475-9e60",
    "note": "",
    "creationDate": "2023-06-28T20:13:20.643567Z",
    "revisionDate": "2023-06-28T20:13:20.643567Z"
  }
]
```

Projekt

Der Projektbefehl wird verwendet, um auf [Projekte](#) zuzugreifen, sie zu manipulieren und zu erstellen. Der Umfang des Zugriffs, der Ihrem Dienstkonto zugewiesen ist, bestimmt, welche Aktionen mit dem **Projekt**-Befehl ausgeführt werden können.

Note

Projekte können von einem Dienstkonto mit schreibgeschütztem Zugriff erstellt werden. Bestehende Projekte, die nicht vom Dienstkonto erstellt wurden, können jedoch nicht ohne **Lese** und **Schreib**zugriff bearbeitet werden.

Projekt erstellen

Verwenden Sie `bws project create`, um ein neues Projekt zu erstellen. Dieser Befehl erfordert einen `NAMEN`.

Bash

```
bws project create <NAME>
```

In diesem Beispiel wird ein Projekt mit dem Namen `Mein Projekt` erstellt.

Bash

```
bws project create "My project"
```

Standardmäßig gibt `bws project create` Objekte als JSON-Array zurück, wie im folgenden Beispiel. Sie können das Ausgabeformat mit der `--output` Flagge ändern ([mehr erfahren](#)).

Bash

```
{
  "object": "project",
  "id": "1c80965c-acb3-486e-ac24-b03000dc7318",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "name": "My project",
  "creationDate": "2023-06-29T13:22:37.942559Z",
  "revisionDate": "2023-06-29T13:22:37.942559Z"
}
```

Projekt löschen

Verwenden Sie `bws project löschen`, um ein oder mehrere durch die `PROJECT_IDS` bezeichnete Projekte zu löschen.

Bash

```
bws project delete <PROJECT_IDS>
```

Für ein einzelnes Projekt, bei dem `f1fe5978-0aa1-4bb0-949b-b03000e0402a` die `PROJEKT_ID` darstellt:

Bash

```
bws project delete f1fe5978-0aa1-4bb0-949b-b03000e0402a
```

Für mehrere Projekte, bei denen `1c80965c-acb3-486e-ac24-b03000dc7318` und `f277fd80-1bd2-4532-94b2-b03000e00c6c` die `PROJEKT_IDS` darstellen:

Bash

```
bws project delete 1c80965c-acb3-486e-ac24-b03000dc7318 f277fd80-1bd2-4532-94b2-b03000e00c6c
```

Ausgabe:

Bash

```
1 project deleted successfully.
```

Projekt bearbeiten

Mit dem Befehl `bearbeiten` können Sie den Namen eines Projekts mit der folgenden Eingabe ändern:

Bash

```
bws project edit <PROJECT_ID> --name <NEW_NAME>
```

Zum Beispiel wird dieser Befehl den Projektnamen in `Mein Projekt 2` ändern.

Bash

```
bws project edit 1c80965c-acb3-486e-ac24-b03000dc7318 --name "My project 2"
```

Standardmäßig gibt `bws Projekt bearbeiten` Objekte als JSON-Array zurück, wie im folgenden Beispiel. Sie können das Ausgabeformat mit der `--output` Flagge ändern ([mehr erfahren](#)).

Bash

```
{
  "object": "project",
  "id": "1c80965c-acb3-486e-ac24-b03000dc7318",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "name": "My project 2",
  "creationDate": "2023-06-29T13:22:37.942559Z",
  "revisionDate": "2023-06-29T13:31:07.927829Z"
}
```

Projekt erhalten

Der `get` Befehl ruft ein spezifisches Projekt ab, auf das das angemeldete Service-Konto aus Ihrem Tresor zugreifen kann. Objekte in Ihrem Tresor, auf die das Servicekonto keinen Zugriff hat, können nicht abgerufen werden.

Bash

```
bws project get <PROJECT_ID>
```

Um ein spezifisches Projekt zu erhalten, verwenden Sie den folgenden Befehl, bei dem `e325ea69-a3ab-4dff-836f-b02e013fe530` eine `PROJECT_ID` darstellt:

Bash

```
bws project get e325ea69-a3ab-4dff-836f-b02e013fe530
```

Standardmäßig gibt `get` Objekte als JSON-Array zurück, wie im folgenden Beispiel. Sie können das Ausgabeformat mit der `--output` Flagge ändern ([erfahren Sie mehr](#)).

Bash

```
{
  "object": "project",
  "id": "e325ea69-a3ab-4dff-836f-b02e013fe530",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "name": "App 1",
  "creationDate": "2023-06-27T19:24:42.181607Z",
  "revisionDate": "2023-06-27T19:24:42.181607Z"
}
```

Projektliste

Um die Projekte aufzulisten, auf die dieses Service-Konto Zugriff hat, verwenden Sie den folgenden Befehl:

Bash

```
bws project list
```

Standardmäßig gibt `Liste` Objekte als JSON-Array zurück, wie im folgenden Beispiel. Sie können das Ausgabeformat mit der `--output` Flagge ändern ([erfahren Sie mehr](#)).

Bash

```
[
  {
    "object": "project",
    "id": "e325ea69-a3ab-4dff-836f-b02e013fe530",
    "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
    "name": "App 1",
    "creationDate": "2023-06-27T19:24:42.181607Z",
    "revisionDate": "2023-06-27T19:24:42.181607Z"
  },
  ...
]
```

Konfiguration

Der Config-Befehl gibt die Servereinstellungen an, die der Secrets Manager CLI verwenden soll. Eine Hauptverwendung von `bws config` besteht darin, die CLI mit einem selbst gehosteten Bitwarden-Server zu verbinden.

Verfügbare Einstellungen beinhalten zum Beispiel `server-basis`, `server-API` und `server-Identität`.

Bash

```
bws config server-base https://my_hosted_server.com
```

Wenn Sie es auf diese Weise tun, werden Ihre angegebenen Serverwerte in einem Standardprofil in einer `~/.bws/config` Datei gespeichert. Sie können die folgenden Optionen verwenden, um alternative Profile und Konfigurationsdateien zu erstellen:

Konfiguration --Profil

Verwenden Sie die Option `--profil` mit dem `config` Befehl, um bestimmte Serverwerte in alternativen Profilen zu speichern, zum Beispiel:

Bash

```
bws config server-base http://other_hosted_server.com --profile dev
```

Einmal erstellt, können Sie dieses Profil mit anderen Befehlen verwenden, um Anfragen an den angegebenen Server zu leiten, zum Beispiel:

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --profile dev
```

config --config-datei

Verwenden Sie die Option `--config-file` mit dem Befehl `config`, um bestimmte Serverwerte in alternativen Konfigurationsdateien zu speichern, zum Beispiel um Werte in einem Standardprofil in einer neuen Konfigurationsdatei zu speichern:

Bash

```
bws config server-base http://third_hosted_server.com --config-file ~/.bws/alt_config
```

Sie können `--config-file` mit `--profile` verketteten, um Werte in alternativen Profilen in alternativen Konfigurationsdateien zu speichern, zum Beispiel:

Bash

```
bws config server-base http://third_hosted_server.com --config-file ~/.bws/alt_config --profile alt_dev
```

Einmal erstellt, können Sie dieses Profil mit anderen Befehlen verwenden, um Anfragen an den angegebenen Server zu leiten, zum Beispiel:

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --config-file ~/.bws/alt_config --profile alt_dev
```

Optionen

-o, --ausgabe

Standardmäßig gibt der Secrets Manager CLI ein JSON-Objekt oder ein Array von JSON-Objekten als Antwort auf Befehle zurück. Das Ausgabeformat kann an Ihre Bedürfnisse angepasst werden, indem Sie die `-o`, `--output` Flagge zusammen mit einer der folgenden Optionen verwenden:

- `json`: Standard. Ausgabe JSON.
- `yaml`: Ausgabe YAML.
- `Tabelle`: Geben Sie eine ASCII-Tabelle aus, mit Schlüsseln als Spaltenüberschriften.
- `tsv`: Ausgabe von Tab-getrennten Werten ohne Schlüssel.
- `keine`: Nur Ausgabefehler und Warnungen.
- `env`: Gebe Geheimnisse im KEY=VALUE Format aus.

Zum Beispiel, der Befehl:

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --output yaml
```

wird das Folgende zurückgeben:

Bash

```
object: secret
id: 2863ced6-eba1-48b4-b5c0-afa30104877a
organizationId: b8824f88-c57c-4a36-8b1a-afa300fe0b52
projectId: 1d0a63e8-3974-4cbd-a7e4-afa30102257e
key: Stripe API Key
value: osiundfpowubefpouweF
note: 'These are notes.'
creationDate: 2023-02-08T15:48:33.470701Z
revisionDate: 2023-02-08T15:48:33.470702Z
```

Note

Bei Verwendung des env-Ausgabeformats wird, wenn der Schlüsselname nicht POSIX-konform ist, dieses Schlüssel-Wert-Paar auskommentiert und am Ende der Ausgabe wird ein Kommentar angezeigt, der darauf hinweist, dass die Ausgabe geändert wurde.

Mit der Verwendung der `--output env` Flagge, zum Beispiel:

Bash

```
bws secret list --output env
```

wird das Folgende zurückgeben:

Bash

```
this_is_a_keyname="this is a key value"
CLOUDFLARE_API_TOKEN="123412341234123412341234"
# This is an invalid keyname="this will get commented-out"

# one or more secrets have been commented-out due to a problematic key name
```

-c, --Farbe

Die Ausgabe kann weiter angepasst werden, indem angegeben wird, ob Sie eine farbige Ausgabe möchten. Verfügbare Werte für diese Option sind `ja`, `nein` und `auto`.

--Zugriffs-Token

Sie können einzelne CLI-Anfragen mit der Option `-t`, `--access-token` für jeden einzelnen Befehl authentifizieren, zum Beispiel:

Bash

```
bws secret list --access-token 0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprlPOYutt  
0:B3h5D+YgLvFiQhWkIq6Bow==
```

--Profil

Verwenden Sie die Option `--profile` mit den Befehlen `list` oder `get`, um anzugeben, welches Profil verwendet werden soll, zum Beispiel:

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --profile dev
```

Beziehen Sie sich auf den `config` Befehl ([hier](#)) um Hilfe beim Verstehen und Einrichten von alternativen Profilen zu erhalten.

--Konfigurationsdatei

Verwenden Sie die Option `--config-file` zusammen mit der Option `--profile` und den Befehlen `list` oder `get`, um anzugeben, welches Profil aus welcher Konfigurationsdatei verwendet werden soll, zum Beispiel:

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --config-file ~/.bws/alt_config --profile alt_d  
ev
```

Beziehen Sie sich auf den `config` Befehl ([hier](#)) um Hilfe beim Verstehen und Einrichten von alternativen Config-Dateien und Profilen zu erhalten.

--server-url

Diese Option kann verwendet werden, um die Server-URL festzulegen, an die die CLI die Anforderung senden wird, die mit einem bestimmten Befehl verknüpft ist, zum Beispiel:

Bash

```
bws list secrets --server-url http://my_hosted_server.com
```

Diese Option überschreibt alle URLs, die über den `config` Befehl konfiguriert wurden (siehe [hier](#)).

--Hilfe

Verwenden Sie diese Option, um Hilfe für einen beliebigen `bws` Befehl auszudrucken.

--Version

Verwenden Sie diese Option, um die Version des `bws` Clients, den Sie verwenden, zu drucken.