

シークレットマネージャー > 開発者ツール

シークレットマネージャー CLI

ヘルプセンターで表示:

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

シークレットマネージャー CLI

シークレットマネージャーのコマンドラインインターフェイス (CLI) は、シークレットを取得し注入するための強力なツールです。シークレットマネージャー CLI は、あなたの保管庫を **作成**、**削除**、**編集**、そしてあなたの秘密とプロジェクトを **リスト** するために使用できます。

シークレットマネージャーの CLI は自己文書化されています。コマンドラインから、以下を使用して利用可能なコマンドについて詳しく学びましょう:

Bash

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

ダウンロードしてインストールしてください

CLI は Windows、macOS、および Linux ディストリビューションでクロスプラットフォームとして使用することができます。シークレットマネージャー CLI をダウンロードしてインストールするには:

<https://github.com/bitwarden/sdk/releases> からシークレットマネージャー CLI をダウンロードしてください。

認証

シークレットマネージャーの CLI は、特定の **サービスアカウント** に対して生成された **アクセストークン** を使用してログインすることができます。これは、サービスアカウントがアクセスできるシークレットとプロジェクトの **みが** CLI を使用して操作できることを意味します。CLI セッションを認証する方法はいくつかあります:

⇒環境変数

あなたは環境変数 **BWS_ACCESS_TOKEN** にアクセストークンの値を保存することで、CLI セッションを認証することができます。例えば:

Bash

```
export BWS_ACCESS_TOKEN=0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprLP0Yutt0:B3h5D+YgLvFiQhWkIq6Bow==
```

⇒インライン

あなたは **-t**、**--access-token** フラグを使用して、個々のコマンドで個々の CLI リクエストを認証することができます。例えば:

Bash

```
bws secret list --access-token 0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprLP0Yutt0:B3h5D+YgLvFiQhWkIq6Bow==
```

⚠ Warning

あなたのワークフローが多くの別々のセッション（アクセストークンを使用して認証する各使用が「セッション」を構成する）を使用して、短時間で同じIPアドレスからリクエストを行う場合、レート制限に遭遇する可能性があります。

コマンド

コマンドは、シークレットマネージャーCLIと対話するために使用されます。シークレットとプロジェクトは、特定のアクセストークンに与えられた権限によって読み取りまたは書き込みが可能です。

シークレットと**プロジェクト**に利用可能なコマンドに関する追加の詳細については、以下を使用してください：

- `bws secret --help`
- `bwsプロジェクト --ヘルプ`

① Note

シークレットマネージャーのバージョン0.3.0から、CLIの構文が変更されました。たとえば、シークレットをリストするコマンドは、`bws list secrets` から `bws secret list` に変更されました。

古い構文は一時的にシークレットマネージャーCLIでサポートされたままになります。

あなたがどのバージョンのシークレットマネージャーCLIを使用しているのかわからない場合は、`bws --version` を入力してください。

秘密

シークレット コマンドは、**シークレット**にアクセスし、操作し、作成するために使用されます。すべてのコマンドと同様に、アクセストークンのアクセス範囲外の**プロジェクト**や秘密は読み取りまたは書き込みができません。

秘密を作る

新しいシークレットを作成するには、`bws secret create` を使用してください。このコマンドは、**キー**、**値**、および **プロジェクトID** が必要です。

Bash

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

必要に応じて、`--note` オプションを使用してメモを追加することができます。例えば：

Bash

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

このコマンドは、デフォルトではJSONオブジェクトを返し、シークレットをシークレットマネージャーに保存します。出力形式は `--output` フラグを使用して変更できます ([詳細を学ぶ](#))。

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"
}
```

秘密の削除

`bws secret 削除` を使用して、`SECRET_IDS` によって指定された1つまたは複数の秘密を削除します。

Bash

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

`id` `be8e0ad8-d545-4017-a55a-b02f014d4158` の単一の秘密を削除するには：

Bash

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

複数の秘密について、`ID` は `382580ab-1368-4e85-bfa3-b02e01400c9f` と `47201c5c-5653-4e14-9007-b02f015b2d82` です：

Bash

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

出力:

Bash

```
1 secret deleted successfully.
```

秘密の編集

秘密を編集するには、以下の構造が選択した値に変更を適用します。CLI からこのコマンドを使用して、シークレット **KEY**、**VALUE**、**NOTE**、または **PROJECT_ID** を編集できます。

Bash

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

例えば、既存の秘密にメモを追加したい場合は：

Bash

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

Note

スペースを含む **メモ** を編集するときは、文字列の周りに引用符を含めてください。

SES_KEY2 が新しい **キー** であり、**0.1982492bc-7f37-4475-9e60** が新しい **値** である複数のフィールドを編集するには：

Bash

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

出力:

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"
}
```

秘密を得る

特定のシークレットを取得するには、`bws secret get` を使用してください。

Bash

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

デフォルトでは、このコマンドは `SECRET_ID` を使用してシークレットオブジェクトを取得します。

Bash

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

デフォルトでは、`get` は次の例に示すように、オブジェクトをJSON配列として返します。出力形式は `--output` フラグを使用して変更できます ([詳細を学ぶ](#))。

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"
}
```

秘密のリスト

サービスアカウントがアクセスできるシークレットをリストするには、次のコマンドを使用します:

Bash

```
bws secret list
```

次のコマンドを使用して、特定のプロジェクト内の秘密だけをリストにすることもできます。ここで、`e325ea69-a3ab-4dff-836f-b02e013fe530` はプロジェクト識別子を表します:

Bash

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

デフォルトでは、`list` は次の例のようにJSON配列としてオブジェクトを返します。出力形式は `--output` フラグを使用して変更できます ([詳細を学ぶ](#))。

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"
  }
]
```

プロジェクト

プロジェクトコマンドは、[プロジェクト](#)にアクセスし、操作し、作成するために使用されます。

あなたのサービスアカウントに割り当てられたアクセスの範囲は、[プロジェクト](#) コマンドで何が完了できるかを決定します。

Note

プロジェクトは、読み取り専用アクセスを持つサービスアカウントによって作成することができます。しかし、サービスアカウントによって作成されていない既存のプロジェクトは、**読み取りと書き込み**のアクセス権がなければ編集することはできません。

プロジェクト作成

新しいプロジェクトを作成するには、`bws project create` を使用してください。このコマンドは `名前` が必要です。

Bash

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

この例では、`マイプロジェクト` という名前のプロジェクトが作成されます。

Bash

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

デフォルトでは、`bws project create` は次の例のように、オブジェクトをJSON配列として返します。出力形式は `--output` フラグを使用して変更できます ([詳しくはこちら](#))。

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"
}
```

プロジェクト削除

`bws project 削除` を使用して、`PROJECT_IDS` で指定された1つまたは複数のプロジェクトを削除します。

Bash

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

単一のプロジェクトについて、`f1fe5978-0aa1-4bb0-949b-b03000e0402a` は `PROJECT_ID` を表します：

Bash

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

複数のプロジェクトにおいて、`1c80965c-acb3-486e-ac24-b03000dc7318` と `f277fd80-1bd2-4532-94b2-b03000e00c6c` が `PROJECT_IDS` を表します：

Bash

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

出力:

Bash

```
1 project deleted successfully.
```

プロジェクトの編集

編集 コマンドを使用すると、次の入力でプロジェクトの名前を変更することができます：

Bash

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

たとえば、このコマンドはプロジェクト名を `マイプロジェクト2` に変更します。

Bash

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

デフォルトでは、**bwsプロジェクト編集** は、次の例のようにオブジェクトをJSON配列として返します。出力形式は `--output` フラグを使用して変更できます ([詳細を学ぶ](#))。

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"
}
```

プロジェクトを取得

`get` コマンドは、ログインしたサービス アカウントがコンテナーからアクセスできる特定のプロジェクトを取得します。サービスアカウントがアクセスできない保管庫内のオブジェクトは取得できません。

Bash

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

特定のプロジェクトを取得するには、e325ea69-a3ab-4dff-836f-b02e013fe530が `PROJECT_ID` を表す次のコマンドを使用します:

Bash

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

デフォルトでは、`get` は次の例のようにJSON配列としてオブジェクトを返します。出力形式は `--output` フラグを使用して変更できます ([詳細を学ぶ](#))。

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"
}
```

プロジェクトリスト

このサービスアカウントがアクセスできるプロジェクトをリストするには、次のコマンドを使用してください：

Bash

```
bws project list
```

デフォルトでは、`list` は次の例のようにJSON配列としてオブジェクトを返します。出力形式は `--output` フラグを使用して変更できます（[詳細を学ぶ](#)）。

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"
  },
  ...
]
```

設定

config コマンドは、シークレットマネージャー CLI が使用するサーバー設定を指定します。`bws config` の主な用途は、CLI をセルフホスト型 Bitwarden サーバに接続することです。

利用可能な設定には、例えば、`server-base`、`server-api`、`server-ID` が含まれます。

Bash

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

この方法で行うと、指定したサーバーの値はデフォルトプロファイルに保存され、`~/.bws/config` ファイルに保存されます。次のオプションを使用して、代替のプロファイルと設定ファイルを作成することができます：

設定 --プロファイル

`--profile` オプションを `config` コマンドと一緒に使用して、指定されたサーバーの値を代替プロファイルに保存します。例えば：

Bash

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

一度作成すると、そのプロファイルを他のコマンドと一緒に使用して、指定されたサーバーへのリクエストをルーティングすることができます。例えば：

Bash

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

config --config-file

`--config-file` オプションを `config` コマンドと一緒に使用して、指定されたサーバー値を代替の設定ファイルに保存します。例えば、新しい設定ファイルのデフォルトプロファイルに値を保存するために使用します。

Bash

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

あなたは `--config-file` と `--profile` を連鎖させて、代替の設定ファイルの代替プロファイルに値を保存することができます。例えば：

Bash

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

一度作成すると、そのプロファイルを他のコマンドと一緒に使用して、指定されたサーバーへのリクエストをルーティングすることができます。例えば：

Bash

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

オプション

-o、--output

デフォルトでは、シークレットマネージャーのCLIは、コマンドに対する応答としてJSONオブジェクトまたはJSONオブジェクトの配列を返します。出力形式は、以下のオプションのいずれかとともに `-o`、`--output` フラグを使用して、あなたのニーズに合わせて変更することができます。

- `json` : デフォルト。JSONを出力します。
- `yaml` : YAMLを出力します。
- `テーブル` : キーを列見出しとしてASCIIテーブルを出力します。
- `tsv` : キーなしでタブ区切りの値を出力します。
- `none` : エラーと警告のみを出力します。
- `env` : 秘密鍵をKEY=VALUE形式で出力します。

例えば、コマンド：

Bash

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

次のものを返します：

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

env出力形式を使用している間、キー名がPOSIX準拠でない場合、そのキー値ペアはコメントアウトされ、出力が変更されたことを示すコメントが出力の最下部に表示されます。

`--output env` フラグを使用して、例えば：

Bash

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

次のものを返します：

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, --カラー

出力は、カラー化した出力を希望するかどうかを指定することでさらにカスタマイズできます。このオプションの利用可能な値は、**はい**、**いいえ**、そして **自動** です。

--アクセストークン

-t、**--access-token** オプションを使用して、個々のコマンドで個々のCLIリクエストを認証できます。例えば：

Bash

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

--プロフィール

--profile オプションを **list** または **get** コマンドと一緒に使用して、使用するプロファイルを指定します。例えば：

Bash

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

代替プロファイルの理解と設定のために、**config** コマンド ([ここ](#)) を参照してください。

--設定ファイル

--config-file オプションを **--profile** オプションと **list** または **get** コマンドと一緒に使用して、どの設定ファイルのどのプロファイルを使用するかを指定します。例えば：

Bash

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

代替の設定ファイルとプロファイルの理解と設定についてのヘルプは、**設定** コマンド ([ここ](#)) を参照してください。

--サーバーURL

このオプションは、CLIが特定のコマンドに関連するリクエストを送信するサーバーURLを設定するために使用できます。例えば：

Bash

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


このオプションは、`config` コマンド ([こちら](#)を参照) で設定されたURLを上書きします。

--ヘルプ

このオプションを使用して、指定された `bws` コマンドのヘルプを印刷します。

--バージョン

このオプションを使用して、使用中の `bws` クライアントのバージョンを印刷します。