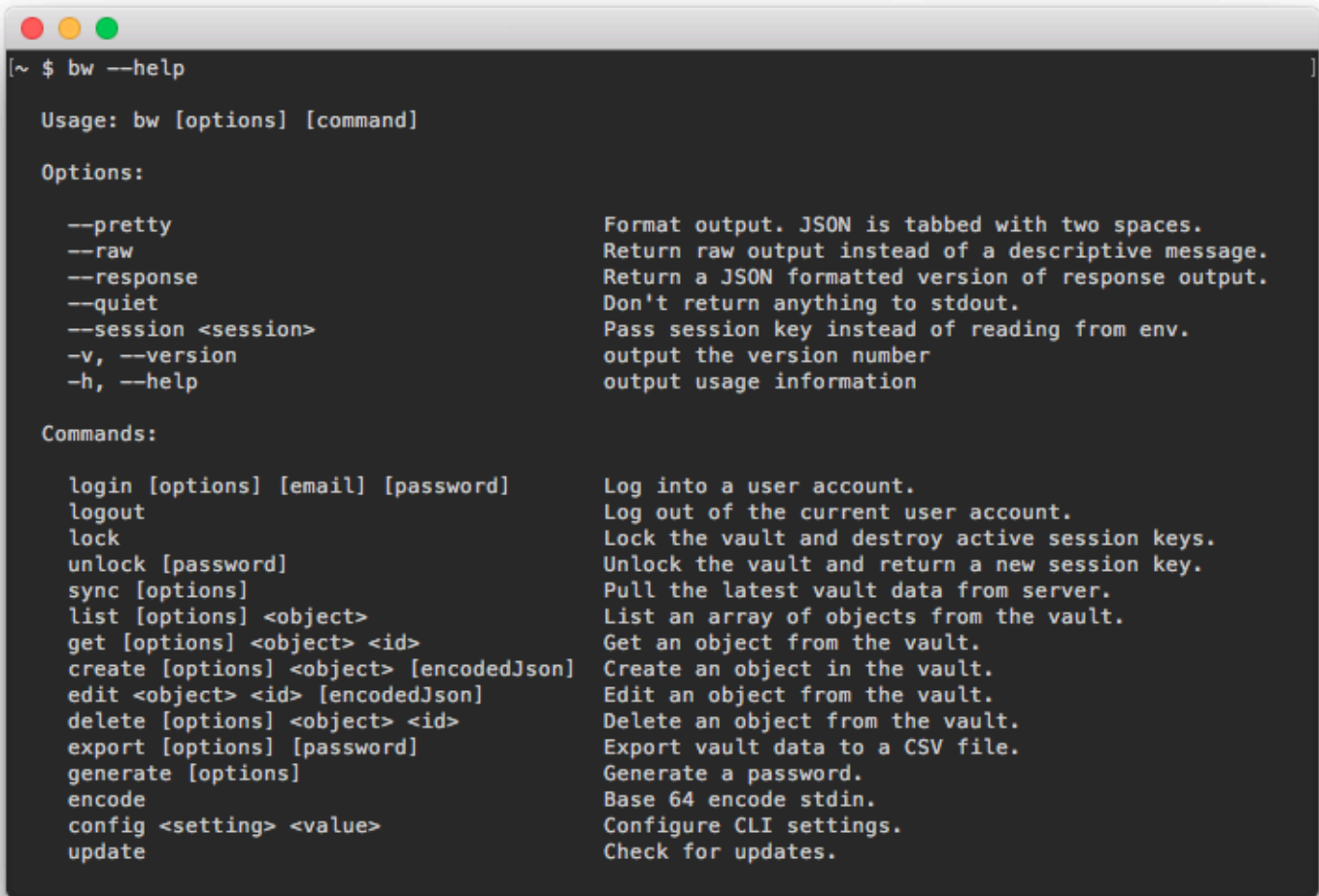


PASSWORD MANAGER > GEREEDSCHAP VOOR ONTWIKKELAARS

Wachtwoordbeheer CLI

Wachtwoordbeheer CLI

De Bitwarden command-line interface (CLI) is een krachtige, volledig uitgeruste tool voor toegang tot en beheer van uw kluis. De meeste functies die u vindt in andere Bitwarden client applicaties (desktop, browserextensie, etc.) zijn beschikbaar via de CLI.



```
[~ $ bw --help]

Usage: bw [options] [command]

Options:

  --pretty          Format output. JSON is tabbed with two spaces.
  --raw            Return raw output instead of a descriptive message.
  --response       Return a JSON formatted version of response output.
  --quiet         Don't return anything to stdout.
  --session <session> Pass session key instead of reading from env.
  -v, --version   output the version number
  -h, --help     output usage information

Commands:

  login [options] [email] [password]  Log into a user account.
  logout                             Log out of the current user account.
  lock                                Lock the vault and destroy active session keys.
  unlock [password]                  Unlock the vault and return a new session key.
  sync [options]                     Pull the latest vault data from server.
  list [options] <object>           List an array of objects from the vault.
  get [options] <object> <id>       Get an object from the vault.
  create [options] <object> [encodedJson] Create an object in the vault.
  edit <object> <id> [encodedJson]  Edit an object from the vault.
  delete [options] <object> <id>    Delete an object from the vault.
  export [options] [password]       Export vault data to a CSV file.
  generate [options]                Generate a password.
  encode                          Base 64 encode stdin.
  config <setting> <value>         Configure CLI settings.
  update                            Check for updates.
```

Bitwarden CLI

De Bitwarden CLI is zelf gedocumenteerd. Leer vanaf de commandoregel de beschikbare commando's kennen met behulp van:

Bash

```
bw --help
```

Of geef `--help` als optie op elk `bw` commando om beschikbare opties en voorbeelden te zien:

Bash

```
bw list --help
```

```
bw move --help
```

De meeste informatie die je nodig hebt kun je vinden met `--help`, maar dit artikel geeft al die informatie weer en gaat dieper in op sommige onderwerpen.

Downloaden en installeren

De CLI kan cross-platform gebruikt worden op Windows, macOS en Linux distributies. Download en installeer de Bitwarden CLI:

Note

Installeer voor arm64-apparaten de CLI met `npm`.

⇒ Oorspronkelijk uitvoerbaar

Volledig verpakte versies van de CLI zijn beschikbaar voor elk platform en hebben geen afhankelijkheden. Download via een van deze links:

- [Windows x64](#)
- [macOS x64](#)
- [Linux x64](#)

Merk op dat, wanneer je de gedownloade native executable gebruikt, je de executable moet toevoegen aan je PATH of anders commando's moet uitvoeren vanuit de map waarnaar het bestand is gedownload.

Tip

Op Linux- en UNIX-systemen krijg je mogelijk een melding `Permission denied`. Als dat het geval is, geef dan toestemming door

Bash

```
chmod +x </path/to/executable>
```

⇒ NPM

Als je Node.js op je systeem hebt geïnstalleerd, kun je de CLI installeren met NPM. Installeren met NPM is de eenvoudigste manier om je installatie up-to-date te houden en zou de **voorkeursmethode moeten zijn voor degenen die al vertrouwd zijn met NPM**:

Bash

```
npm install -g @bitwarden/cli
```

Bekijk het pakket op npmjs.org.

Note

Voor het installeren van de Bitwarden CLI op Linux-systemen met **npm** moet mogelijk eerst de **build-essential** afhankelijkheid (of distributie-equivalent) worden geïnstalleerd. Bijvoorbeeld:

Plain Text

```
apt install build-essential
```

⇒Chocolatey

Installeren met Chocolatey:

Bash

```
choco install bitwarden-cli
```

Bekijk het pakket op community.chocolatey.org.

⇒Snap

Installeren met snap:

Bash

```
sudo snap install bw
```

Bekijk het pakket op snapcraft.io.

Inloggen

Voordat je inlogt, moet je ervoor zorgen dat je CLI is verbonden met de juiste server (bijvoorbeeld [EU cloud](#) of zelf gehost) met het config commando([meer informatie](#)). Er zijn drie methoden om in te loggen in de Bitwarden CLI met het commando **login**, die elk geschikt zijn voor verschillende situaties. Bekijk de volgende opties om te bepalen welke methode je moet gebruiken:

- [E-mail en hoofdwachtwoord gebruiken](#)
- [Een API-sleutel gebruiken](#)
- [SSO gebruiken](#)

Welke optie je ook gebruikt, gebruik altijd de commando's **bw lock** of **bw logout** als je klaar bent.

 **Tip**

Aanmelden met e-mail en hoofdwachtwoord maakt gebruik van je hoofdwachtwoord en kan daarom de opdrachten voor **aanmelden** en **ontgrendelen** samenvoegen om je identiteit te verifiëren en je kluis tegelijkertijd te ontsleutelen. Als je een **API sleutel** of **SSO** gebruikt, moet je het **login** commando opvolgen met een expliciete **bw unlock** als je direct met vault data gaat werken.

Je hoofdwachtwoord is namelijk de bron van de sleutel die nodig is om kluisgegevens te ontsleutelen. Er zijn echter een paar commando's waarbij je kluis niet ontsleuteld hoeft te worden, zoals **config**, **encode**, **generate**, **update** en **status**.

E-mail en wachtwoord gebruiken

Inloggen met e-mail en wachtwoord wordt **aanbevolen voor interactieve sessies**. Om in te loggen met e-mail en wachtwoord:

Bash

```
bw login
```

Er wordt dan gevraagd om je **e-mailadres**, **hoofdwachtwoord** en (indien ingeschakeld) **de code voor tweestapsaanmelding**. De CLI ondersteunt momenteel tweestapslogin via **authenticator**, **e-mail** of **Yubikey**.

U kunt deze factoren aan elkaar rijgen in een enkel commando zoals in het volgende voorbeeld, maar dit wordt om veiligheidsredenen niet aangeraden:

Bash

```
bw login [email] [password] --method <method> --code <code>
```

Zie [Enums](#) voor login in twee stappen waarden.

 **Tip**

Wordt u gevraagd om aanvullende verificatie of krijgt u een **Uw verificatieverzoek lijkt afkomstig te zijn van een bot**. foutmelding? Gebruik je API Key **client_secret** om de authenticatie-uitdaging te beantwoorden. [Meer informatie](#).

Een API-sleutel gebruiken

Inloggen met de **persoonlijke API-sleutel** wordt aanbevolen voor geautomatiseerde workflows, voor het verlenen van toegang tot een externe applicatie of als je account een 2FA-methode gebruikt die niet wordt ondersteund door de CLI (FIDO2 of Duo). Om in te loggen met de API-sleutel:

Bash

```
bw login --apikey
```

Er wordt dan gevraagd om je persoonlijke **client_id** en **client_secret**. Zodra je sessie is geverifieerd met deze waarden, kun je het **unlock** commando gebruiken. [Meer informatie](#).

Tip

Als uw organisatie [SSO vereist](#), kunt u nog steeds `--apikey` gebruiken om in te loggen op de CLI.

API-sleutel omgevingsvariabelen gebruiken

In scenario's waar geautomatiseerd wordt gewerkt met de Bitwarden CLI, kunt u omgevingsvariabelen opslaan om te voorkomen dat u handmatig moet ingrijpen bij de authenticatie.

Naam omgevingsvariabele	Vereiste waarde
BW_CLIENTID	klant_id
BW_CLIËNTGEHEIM	cliënt_geheim

SSO gebruiken

Aanmelden met [SSO](#) wordt aanbevolen als een organisatie SSO-verificatie vereist. Om in te loggen met SSO:

Bash

```
bw login --sso
```

Dit zal de [SSO-authenticatiestroom](#) in je webbrowser starten. Zodra je sessie is geverifieerd, kun je het `unlock` commando gebruiken. [Meer informatie](#).

Tip

Als uw organisatie [SSO vereist](#), kunt u ook `--apikey` gebruiken om in te loggen op de CLI.

Aanmelden bij meerdere accounts

Net als bij het gebruik van [accountwisseling](#) op andere Bitwarden-apps, heeft de CLI de mogelijkheid om in te loggen op meerdere accounts tegelijk met behulp van de omgevingsvariabele `BITWARDENCLI_APPDATA_DIR` die verwijst naar de locatie van een `bw-configuratiebestand`, meestal `data.json` genaamd. Je kunt bijvoorbeeld aliassen instellen in een `.bashrc` profiel voor twee verschillende configuraties:

Bash

```
alias bw-personal="BITWARDENCLI_APPDATA_DIR=~/.config/Bitwarden\ CLI\ Personal /path/to/bw $@"
alias bw-work="BITWARDENCLI_APPDATA_DIR=~/.config/Bitwarden\ CLI\ Work /path/to/bw $@"
```

Met dit voorbeeld zou je dan kunnen inloggen op twee accounts door eerst `source /path/to/.bashrc` uit te voeren, gevolgd door `bw-personal login` en `bw-work login`.

Ontgrendel

Als je een API sleutel of SSO gebruikt om in te loggen, moet je het `login` commando opvolgen met een expliciete `bw unlock` als je direct met vault data werkt.

Het ontgrendelen van uw kluis genereert een **sessiesleutel** die fungeert als ontcijferingssleutel voor interactie met gegevens in uw kluis. De **sessiesleutel moet gebruikt worden** om elk commando uit te voeren dat kluisgegevens aanraakt (bijvoorbeeld, `list`, `get`, `edit`). Sessiesleutels zijn geldig totdat ze ongeldig worden gemaakt met `bw lock` of `bw logout`, maar ze blijven niet bestaan als je een nieuw terminalvenster opent. Genereer op elk moment een nieuwe sessiesleutel met:

```
Bash
```

```
bw unlock
```

Als je klaar bent, beëindig dan altijd je sessie met het commando `bw lock`.

Opties ontgrendelen

U kunt de opties `--passwordenv` of `--passwordfile` gebruiken met `bw unlock` om bijvoorbeeld uw hoofdwachtwoord op te halen in plaats van het handmatig in te voeren:

1. Het volgende zoekt naar een omgevingsvariabele `BW_PASSWORD`. Als `BW_PASSWORD` niet leeg is en de juiste waarden heeft, zal de CLI met succes ontgrendelen en een sessiesleutel teruggeven:

```
Bash
```

```
bw unlock --passwordenv BW_PASSWORD
```

2. Het volgende zoekt naar het bestand `~/Users/Me/Documents/mp.txt` (dat uw hoofdwachtwoord als eerste regel moet hebben). Als het bestand niet leeg is en een correcte waarde heeft, zal de CLI met succes ontgrendelen en een sessiesleutel teruggeven:

```
Bash
```

```
bw unlock --passwordfile ~/Users/Me/Documents/mp.txt
```

Warning

Als je de `--passwordfile` optie gebruikt, bescherm dan je wachtwoordbestand door de toegang te beperken tot alleen de gebruiker die `bw unlock` uit moet voeren en geef alleen leestoegang aan die gebruiker.

Een sessiesleutel gebruiken

Wanneer je je kluis ontgrendelt met `bw login` met e-mail en wachtwoord of `bw unlock`, zal de CLI zowel een `export BW_SESSION` (Bash) als een `env: BW_SESSION` (PowerShell) commando teruggeven, inclusief je sessiesleutel. Kopieer en plak de relevante invoer om de

vereiste omgevingsvariabele op te slaan.

Als de `BW_SESSION` omgevingsvariabele is ingesteld, zullen `bw` commando's naar die variabele verwijzen en kunnen ze netjes worden uitgevoerd, bijvoorbeeld:

Bash

```
export BW_SESSION="5PBYGU+5yt3RHcCjoeJKx/wByU34vokGRZjXpSH7Ylo8w=="  
  
bw list items
```

Als alternatief, als je de omgevingsvariabele niet instelt, kun je de sessiesleutel als optie doorgeven bij elk `bw` commando:

Bash

```
bw list items --session "5PBYGU+5yt3RHcCjoeJKx/wByU34vokGRZjXpSH7Ylo8w=="
```

Tip

Als je klaar bent, beëindig dan altijd je sessie met de commando's `bw lock` of `bw logout`. Hierdoor wordt de actieve sessiesleutel ongeldig gemaakt.

Kernopdrachten

maken

Het commando `aanmaken` maakt een nieuw object (`item`, `bijlage` en meer) aan in je kluis:

Bash

```
bw create (item|attachment|folder|org-collection) <encodedJson> [options]
```

De opdracht `create` gebruikt gecodeerde JSON. Een typische workflow voor het maken van een object zou er ongeveer zo uit kunnen zien:

1. Gebruik het `get template` commando (zie [get core commando's](#) voor details) om het juiste JSON sjabloon voor het objecttype uit te voeren.
2. Gebruik een [commandoregel JSON-processor](#) zoals `jq` om de uitgevoerde sjabloon naar wens te manipuleren.
3. Gebruik het `encode` commando (zie [details](#)) om de gemanipuleerde JSON te coderen.
4. Gebruik de opdracht `create` om een object te maken van de gecodeerde JSON.

Bijvoorbeeld:

Bash

```
bw get template folder | jq '.name="My First Folder"' | bw encode | bw create folder
```

of

Bash

```
bw get template item | jq ".name=\"My Login Item\" | .login=$(bw get template item.login | jq '.use  
rname=\"jdoe\" | .password=\"myp@ssword123\"')\" | bw encode | bw create item
```

Bij succesvolle creatie wordt het nieuw aangemaakte object geretourneerd als JSON.

andere itemtypes maken

Het create commando maakt standaard een login item aan, maar je kunt een [command-line JSON processor](#) zoals jq gebruiken om een `.type=` attribuut te veranderen om andere [itemtypes](#) aan te maken:

Naam	Waarde
Inloggen	<code>.type=1</code>
Beveiligde notitie	<code>.type=2</code>
Kaart	<code>.type=3</code>
Identiteit	<code>.type=4</code>

Het volgende commando maakt bijvoorbeeld een beveiligde notitie:

Bash

```
bw get template item | jq '.type = 2 | .secureNote.type = 0 | .notes = "Contents of my Secure Not  
e." | .name = "My Secure Note"' | bw encode | bw create item
```

Note

Merk in het bovenstaande voorbeeld op dat beveiligde notities een subtemplate (`.secureNote.type`) vereisen. Je kunt subjablonen van itemtypes bekijken met `bw get template` (zie [hier](#) voor details).

bijlage maken

De opdracht `Maak bijlage` voegt een bestand toe aan een `bestand` item.

In tegenstelling tot andere `aanmaakbewerkingen`, hoeft u geen JSON-verwerker of `codering` te gebruiken om een bijlage aan te maken. Gebruik in plaats daarvan de optie `--bestand` om het bij te voegen bestand op te geven en de optie `--itemid` om het bij te voegen item op te geven. Bijvoorbeeld:

Bash

```
bw create attachment --file ./path/to/file --itemid 16b15b89-65b3-4639-ad2a-95052a6d8f66
```

Tip

Als je het exacte `itemid` dat je wilt gebruiken niet weet, gebruik dan `bw get item` om het item terug te geven (zie [details](#)), inclusief het `id`.

krijgen

Het `get` commando haalt een enkel object (`item`, `gebruikersnaam`, `wachtwoord` en meer) op uit je kluis:

Bash

```
bw get (item|username|password|uri|totp|exposed|attachment|folder|collection|organization|org-collection|template|fingerprint) <id> [options]
```

Het `get` commando neemt een item `id` of string als argument. Als je een tekenreeks gebruikt (bijvoorbeeld iets anders dan een exact `id`), zal `get` in je kluisobjecten zoeken naar een object met een waarde die overeenkomt. Het volgende commando zou bijvoorbeeld een Github-wachtwoord opleveren:

Bash

```
bw get password Github
```

Note

Het `get` commando kan **maar één resultaat teruggeven**, dus je moet specifieke zoektermen gebruiken. Als er meerdere resultaten worden gevonden, geeft de CLI een foutmelding.

bijlage krijgen

De opdracht `get attachment` downloadt een bestandsbijlage:

Bash

```
bw get attachment <filename> --itemid <id>
```

Het `get attachment` commando neemt een **bestandsnaam** en **exacte id**. Standaard zal `get attachment` de bijlage downloaden naar de huidige werkmapp. Je kunt de optie `--output` gebruiken om bijvoorbeeld een andere uitvoermap op te geven:

Bash

```
bw get attachment photo.png --itemid 99ee88d2-6046-4ea7-92c2-acac464b1412 --output /Users/myaccount/Pictures/
```

Note

Bij gebruik van `--output` **moet** het pad eindigen met een forward slash (/) om een map of een bestandsnaam te specificeren (/Users/mijnaccount/Pictures/foto.png).

notities krijgen

Het `get notes` commando haalt de notitie op voor elk item in de kluis:

Bash

```
bw get notes <id>
```

Het `get notes` commando neemt een exact item **id** of string. Als je een tekenreeks gebruikt (bijvoorbeeld iets anders dan een exact **id**), zal `get notes` in je kluisobjecten zoeken naar een object met een waarde die overeenkomt. Het volgende commando zou bijvoorbeeld een Github-notitie opleveren:

Bash

```
bw get notes Github
```

sjabloon krijgen

De opdracht `get template` retourneert de verwachte JSON-opmaak voor een object (`item`, `item.field`, `item.login` en meer):

Bash

```
bw get template (item|item.field|item.login|item.login.uri|item.card|item.identity|item.securenote|folder|collection|item-collections|org-collection)
```

Hoewel je `get template` kunt gebruiken om het formaat naar je scherm te exporteren, is de meest voorkomende toepassing om de output in een `bw create` operatie te pipen, met behulp van een `commandoregel JSON processor` zoals `jq` en `bw encode` om de waarden die zijn opgehaald uit de template te manipuleren, bijvoorbeeld:

Bash

```
bw get template folder | jq '.name="My First Folder"' | bw encode | bw create folder
```

Note

Elk `item.xxx` sjabloon moet worden gebruikt als een subobject van een `item` sjabloon, bijvoorbeeld:

Bash

```
bw get template item | jq ".name=\"My Login Item\" | .login=$(bw get template item.login | jq  
' .username=\"jdoe\" | .password=\"myp@ssword123\"')\" | bw encode | bw create item
```

bewerken

Het `bewerken` commando bewerkt een object (`item`, `itemverzamelingen`, etc.) in je kluis:

Bash

```
bw edit (item|item-collections|folder|org-collection) <id> [encodedJson] [options]
```

Het `bewerken` commando neemt een **exact id** (het object dat bewerkt moet worden) en gecodeerde JSON (bewerkingen die gedaan moeten worden). Een typische workflow zou er ongeveer zo uit kunnen zien:

1. Gebruik het `get` commando (zie [details](#)) om het te bewerken object uit te voeren.
2. Gebruik een `opdrachtregel JSON-processor` zoals `jq` om het uitvoerobject naar wens te manipuleren.
3. Gebruik het `encode` commando (zie [details](#)) om de gemanipuleerde JSON te coderen.
4. Gebruik het commando `Bewerken` (inclusief object `id`) om het object te bewerken.

Bijvoorbeeld om het wachtwoord van een aanmeldingsitem te bewerken:

Bash

```
bw get item 7ac9cae8-5067-4faf-b6ab-acfd00e2c328 | jq '.login.password="newp@ssw0rd"' | bw encode |  
bw edit item 7ac9cae8-5067-4faf-b6ab-acfd00e2c328
```

Of om de collectie(s) waar een item in zit te bewerken:

Bash

```
echo '["5c926f4f-de9c-449b-8d5f-aec1011c48f6"]' | bw encode | bw edit item-collections 28399a57-73a0-45a3-80f8-aec1011c48f6 --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5
```

Of om een collectie te bewerken:

Bash

```
bw get org-collection ee9f9dc2-ec29-4b7f-9afb-aac8010631a1 --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5 | jq '.name="My Collection"' | bw encode | bw edit org-collection ee9f9dc2-ec29-4b7f-9afb-aac8010631a1 --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5
```

De opdracht **Bewerken** voert een **vervangbewerking** uit op het object. Na voltooiing wordt het bijgewerkte object geretourneerd als JSON.

lijst

Het **list** commando haalt een reeks objecten (**items**, **mappen**, **collecties** en meer) op uit je kluis:

Bash

```
bw list (items|folders|collections|organizations|org-collections|org-members) [options]
```

Opties voor het **list** commando zijn **filters** die worden gebruikt om te bepalen wat er wordt geretourneerd, inclusief **--url**, **--folderid**, **--collectionid**, **--organizationid** en **--trash**. Elk filter accepteert **null** of **notnull**. Door meerdere filters in één opdracht te combineren, wordt bijvoorbeeld een OR-bewerking uitgevoerd:

Bash

```
bw list items --folderid null --collectionid null
```

Dit commando geeft items terug die niet in een map of verzameling zitten.

Daarnaast kun je naar specifieke objecten **zoeken** met **--search**. Door filter en zoeken in één opdracht te combineren, wordt bijvoorbeeld een EN-bewerking uitgevoerd:

Bash

```
bw list items --search github --folderid 9742101e-68b8-4a07-b5b1-9578b5f88e6f
```

Deze opdracht zoekt naar items met de string **github** in de opgegeven map.

verwijderen

Het `delete` commando verwijdert een object uit je kluis. `delete` heeft **alleen een exact id** als argument.

Bash

```
bw delete (item|attachment|folder|org-collection) <id> [options]
```

Standaard stuurt `verwijderen` een item naar de [Prullenbak](#), waar het 30 dagen blijft staan. Je kunt een item permanent verwijderen met de optie `-p`, `--permanent`.

Bash

```
bw delete item 7063feab-4b10-472e-b64c-785e2b870b92 --permanent
```

Om een `org-verzameling` te verwijderen, moet je ook `--organizationid` opgeven. Zie [Organisatie-ID's](#).

Warning

Terwijl items die verwijderd zijn met `delete` hersteld kunnen worden met het `restore` commando voor maximaal 30 dagen (zie [details](#)), zijn items die verwijderd zijn met `delete --permanent` **volledig verwijderd en onherstelbaar**.

herstellen

Het commando `restore` herstelt een verwijderd object uit je prullenbak. `restore` neemt **alleen een exacte id** als argument.

Bash

```
bw restore (item) <id> [options]
```

Bijvoorbeeld:

Bash

```
bw restore item 7063feab-4b10-472e-b64c-785e2b870b92
```

verzenden

Het commando `send` maakt een [Bitwarden Send-object](#) voor kortstondig delen. In deze sectie worden eenvoudige `verzendsbewerkingen` beschreven, maar `send` is een zeer flexibele tool en we raden aan om het speciale artikel over [Send from CLI](#) te raadplegen.

Een eenvoudige tekst maken Send:

Bash

```
bw send -n "My First Send" -d 7 --hidden "The contents of my first text Send."
```

Een eenvoudig bestand maken Send:

Bash

```
bw send -n "A Sensitive File" -d 14 -f /Users/my_account/Documents/sensitive_file.pdf
```

ontvangen

Het **ontvangstcommando** geeft toegang tot een Bitwarden Send-object. Om een Send-object te ontvangen:

Bash

```
bw receive --password passwordforaccess https://vault.bitwarden.com/#/send/yawoill8rk6VM6zCATXv2A/9WN8wD-hzsDJjfnXLeNc2Q
```

Opdrachten voor organisaties

Organisatie-ID's

Om toegang te krijgen tot een organisatie via de CLI heb je kennis nodig van een ID voor je organisatie en ID's voor individuele leden en collecties.

Vraag deze informatie rechtstreeks op via de CLI met bijvoorbeeld **bw list**:

Bash

```
bw list organizations
bw list org-members --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5
bw list org-collections --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5
```

💡 Tip

Je kunt zowel **collecties** als **org-collecties** in een lijst opnemen. Het **bw list collections** commando zal alle collecties tonen, onafhankelijk van bij welke organisatie ze horen. **bw list org-collections** zal alleen collecties tonen die bij de organisatie horen die is opgegeven met **--organizationid**.

verplaatsen

📌 Note

Augustus 2021: De **share-opdracht** is gewijzigd in **move**. Lees [meer](#).

Het **move** commando verplaatst een kluisitem naar een organisatie:

Bash

```
bw move <itemid> <organizationid> [encodedJson]
```

Het **move** commando vereist dat je een collection ID **codeert** en neemt een **exact id** (het object dat je wilt delen) en een **exact organizationid** (de organisatie waar je het object aan wilt delen). Bijvoorbeeld:

Bash

```
echo '["bq209461-4129-4b8d-b760-acd401474va2"]' | bw encode | bw move ed42f44c-f81f-48de-a123-ad01013132ca dfghbc921-04eb-43a7-84b1-ac74013bqb2e
```

Na voltooiing wordt het bijgewerkte item geretourneerd.

bevestigen

Het **confirm** commando bevestigt **uitgenodigde leden** die hun uitnodiging hebben geaccepteerd:

Bash

```
bw confirm org-member <id> --organizationid <orgid>
```

De opdracht **confirm** neemt bijvoorbeeld een **exacte member id** en een **exacte organizationID**:

Bash

```
bw confirm org-member 7063feab-4b10-472e-b64c-785e2b870b92 --organizationid 310d5ffd-e9a2-4451-af87-ea054dce0f78
```

Andere opdrachten

config

Het **config** commando specificeert de instellingen die de Bitwarden CLI moet gebruiken:

Bash

```
bw config server <setting> [value]
```

Het belangrijkste gebruik van **bw config** is om je CLI te verbinden met een zelf gehoste Bitwarden-server:

Bash

```
bw config server https://your.bw.domain.com
```

 **Tip**

Maak verbinding met de Bitwarden [EU-server](#) door het volgende commando uit te voeren:

Bash

```
bw config server https://vault.bitwarden.eu
```

Geef `bw config server` door zonder waarde om de server te lezen waarmee je verbonden bent.

Gebruikers met unieke instellingen kunnen ervoor kiezen om de URL van elke service afzonderlijk op te geven. Merk op dat elk volgend gebruik van het config commando alle vorige specificaties zal overschrijven, dus dit moet als een enkel commando worden uitgevoerd elke keer dat je een verandering aanbrengt:

Bash

```
bw config server --web-vault <url> \  
  --api <url> \  
  --identity <url> \  
  --icons <url> \  
  --notifications <url> \  
  --events <url> \  
  --key-connector <url>
```

 **Note**

Het commando `bw config server --key-connector` is vereist als je organisatie [Key Connector](#) gebruikt en je de optie `--api key` gebruikt om in te loggen nadat je je hoofdwachtwoord hebt verwijderd.

Neem contact op met de eigenaar van een organisatie om de vereiste URL te krijgen.

sync

De [synchronisatieopdracht](#) downloadt uw versleutelde kluis van de Bitwarden-server. Dit commando is vooral handig als u iets hebt veranderd in uw Bitwarden-kluis in een andere clienttoepassing (bijvoorbeeld webkluis, browserextensie, mobiele app) sinds [het inloggen](#) via de CLI.

Bash

```
bw sync
```

Je kunt de optie `--last` meegeven om alleen het tijdstempel(ISO 8601) van de laatste keer dat er gesynchroniseerd is terug te geven.

💡 Tip

Het is belangrijk om te weten dat `sync` alleen een pull vanaf de server **uitvoert**. Gegevens worden automatisch naar de server gepushed telkens wanneer je een wijziging aanbrengt in je kluis (bijvoorbeeld **aanmaken**, **bewerken**, **verwijderen**).

coderen

De **encodeeropdracht** Base 64 codeert stdin. Dit commando wordt meestal gebruikt in combinatie met een **commandoregel JSON processor** zoals `jq` bij het uitvoeren van bijvoorbeeld **create** en **edit** operaties:

Bash

```
bw get template folder | jq '.name="My First Folder"' | bw encode | bw create folder

bw get item 7ac9cae8-5067-4faf-b6ab-acfd00e2c328 | jq '.login.password="newp@ssw0rd"' | bw encode |
bw edit item 7ac9cae8-5067-4faf-b6ab-acfd00e2c328
```

importeer

De opdracht **importeren** importeert gegevens uit een Bitwarden-export of een andere **ondersteunde toepassing voor wachtwoordbeheer**. Het commando moet naar een bestand wijzen en de volgende argumenten bevatten:

Bash

```
bw import <format> <path>
```

Bijvoorbeeld:

Bash

```
bw import lastpasscsv /Users/myaccount/Documents/mydata.csv
```

 **Tip**

Bitwarden ondersteunt veel formaten voor import, te veel om hier op te noemen! Gebruik `bw import --formats` om de lijst terug te geven in je CLI, of [kijk hier](#).

Als u een gecodeerd .json-bestand importeert [dat u met een wachtwoord hebt aangemaakt](#), wordt u gevraagd het wachtwoord in te voeren voordat het importeren wordt voltooid.

export

Het commando `exporteren` exporteert kluisgegevens als een `.json-` of `.csv-bestand` of als een `gecodeerd json-bestand`:

Bash

```
bw export [--output <filePath>] [--format <format>] [--password <password>] [--organizationid <orgid>]
```

Standaard genereert het `export` commando een `.csv` (gelijk aan het opgeven van `--format csv`) naar de huidige werkmapp, maar je kunt dit ook opgeven:

- `--format json` om een `.json` bestand te exporteren
- `--format encrypted_json` om een `versleuteld json-bestand` te exporteren
 - `--password` om een wachtwoord op te geven om te gebruiken om `encrypted_json` exports te versleutelen in plaats van je `account encryptiesleutel`
- `--output` om te exporteren naar een specifieke locatie
- `--raw` om de export terug te sturen naar stdout in plaats van naar een bestand

exporteren vanuit een organisatiekluis

Met het commando `exporteren` met de optie `--organizationid` kun je een organisatiekluis exporteren:

Bash

```
bw export --organizationid 7063feab-4b10-472e-b64c-785e2b870b92 --format json --output /Users/myaccount/Downloads/
```

genereren

Het commando `genereren` genereert een sterk wachtwoord of `wachtwoordzin`:

Bash

```
bw generate [--lowercase --uppercase --number --special --length <length> --passphrase --separator <separator> --words <words>]
```

Standaard genereert het commando **genereren** een wachtwoord van 14 tekens met hoofdletters, kleine letters en cijfers. Dit is het equivalent van slagen:

Bash

```
bw generate -uln --length 14
```

Je kunt complexere wachtwoorden genereren met de opties die beschikbaar zijn voor het commando, waaronder:

- **--uppercase, -u** (inclusief hoofdletters)
- **-kleine letters, -l** (inclusief kleine letters)
- **--nummer, -n** (nummers opnemen)
- **--special, -s** (speciale tekens opnemen)
- **--lengte** (lengte van het wachtwoord, min. 5)

een wachtwoordzin genereren

Met het commando **genereren** met de optie **--passphrase** kun je een wachtwoordzin genereren in plaats van een wachtwoord:

Bash

```
bw generate --passphrase --words <words> --separator <separator>
```

Standaard genereert **bw generate --passphrase** een wachtwoordzin van drie woorden, gescheiden door een streepje (-). Dit is het equivalent van slagen:

Bash

```
bw generate --passphrase --words 3 --separator -
```

Je kunt een complexe wachtwoordzin genereren met de opties die beschikbaar zijn voor het commando, waaronder:

- **--words** (aantal woorden)
- **--separator** (scheidingstekens)

- `--hoofdletter`, `-c` (ook om de passphrase een hoofdletter te geven)
- `--includeNumber` (nummers opnemen in de wachtwoordzin)

update

Het `update` commando controleert of je Bitwarden CLI de meest recente versie gebruikt. `update` `update` de CLI niet automatisch voor je.

Bash

```
bw update
```

Als er een nieuwe versie wordt gedetecteerd, moet je de nieuwe versie van de CLI downloaden met behulp van de afgedrukte URL voor het uitvoerbare bestand, of met behulp van de hulpmiddelen die beschikbaar zijn voor de pakketbeheerder die je hebt gebruikt om de CLI te downloaden (bijvoorbeeld `npm install -g @bitwarden/cli`).

status

Het commando `status` geeft statusinformatie over de Bitwarden CLI, inclusief de geconfigureerde server URL, het tijdstempel voor de laatste synchronisatie (ISO 8601), het e-mailadres en de ID van de gebruiker en de status van de kluis.

Bash

```
bw status
```

Status retourneert informatie als een JSON-object, bijvoorbeeld:

Bash

```
{
  "serverUrl": "https://bitwarden.example.com",
  "lastSync": "2020-06-16T06:33:51.419Z",
  "userEmail": "user@example.com",
  "userId": "00000000-0000-0000-0000-000000000000",
  "status": "unlocked"
}
```

`status` kan een van de volgende zijn:

- `"unlocked"`, wat aangeeft dat je ingelogd bent en dat je kluis ontgrendeld is (een `BW_SESSION` key omgevingsvariabele wordt opgeslagen met een actieve sessiesleutel)
- `"locked"`, geeft aan dat je bent ingelogd maar dat je kluis is vergrendeld (er wordt geen omgevingsvariabele `BW_SESSION` key opgeslagen met een actieve sessiesleutel)

- "niet-geauthenticeerd", wat aangeeft dat je niet bent ingelogd



Wanneer "status": "niet-geauthenticeerd", zullen `lastSync`, `userEmail` en `userID` altijd `null` teruggeven.

serveer

Het `serve` commando start een lokale express webserver die gebruikt kan worden om alle acties uit te voeren die toegankelijk zijn vanuit de CLI in de vorm van RESTful API calls vanuit een HTTP interface.

Bash

```
bw serve --port <port> --hostname <hostname>
```

Standaard zal `serve` de webserver starten op poort 8087, maar je kunt een andere poort opgeven met de `--port` optie.

Standaard zal `serve` je API webserver binden aan `localhost`, maar je kunt een alternatieve hostnaam opgeven met de `--hostname` optie. API-verzoeken kunnen alleen worden gedaan vanaf de gebonden hostnaam.

Standaard blokkeert `serve` elk verzoek met een `Origin` header. Je kunt deze bescherming omzeilen met de optie `--disable-origin-protection`, maar **dit wordt niet aangeraden**.

Warning

Je kunt `--hostname all` opgeven voor geen hostname binding, maar dan kan elke machine op het netwerk API verzoeken doen.

Bekijk de [API specs](#) voor hulp bij het maken van aanroepen met `serve`.

Bijlagen

Globale opties

De volgende opties zijn globaal beschikbaar:

Optie	Beschrijving
<code>--pretty</code>	Formaat uitvoer. JSON is getabd met twee spaties.
<code>--ruw</code>	Geef ruwe uitvoer in plaats van een beschrijvend bericht.
<code>--antwoord</code>	Geef een JSON-geformatteerde versie van de responsuitvoer.

Optie	Beschrijving
<code>--Stille</code>	Stuur niets terug naar stdout. Je zou deze optie bijvoorbeeld kunnen gebruiken als je een credentiewaarde naar een bestand of applicatie piped.
<code>--Geen interactie</code>	Vraag niet om interactieve gebruikersinvoer.
<code>--sessie</code>	Sessiesleutel doorgeven in plaats van lezen van een omgevingsvariabele.
<code>-v, --versie</code>	Geef het versienummer van de Bitwarden CLI weer.
<code>-h, --help</code>	Help-tekst voor de opdracht weergeven.

ZSH-shell voltooiing

De Bitwarden CLI biedt ondersteuning voor ZSH shell completion. Gebruik een van de volgende methoden om shell voltooiing in te stellen:

1. **Vanilla ZSH:** Voeg de volgende regel toe aan je `.zshrc` bestand:

Bash

```
eval "$(bw completion --shell zsh); compdef _bw bw;"
```

2. **Vanilla (vendor-completions):** Voer de volgende opdracht uit:

Bash

```
bw completion --shell zsh | sudo tee /usr/share/zsh/vendor-completions/_bw
```

3. **zinit:** Voer de volgende commando's uit:

Bash

```
bw completion --shell zsh > ~/.local/share/zsh/completions/_bw
zinit creinstall ~/.local/share/zsh/completions
```

Zelfondertekende certificaten gebruiken

Als uw zelf gehoste Bitwarden-server een zelfondertekend TLS-certificaat beschikbaar stelt, geef dan de Node.js-omgevingsvariabele `NODE_EXTRA_CA_CERTS` op:

Bash

Bash

```
export NODE_EXTRA_CA_CERTS="absolute/path/to/your/certificates.pem"
```

PowerShell

Bash

```
$env:NODE_EXTRA_CA_CERTS="absolute/path/to/your/certificates.pem"
```

Enums

De volgende tabellen geven een opsomming van waarden die vereist zijn in gedocumenteerde scenario's:

Inlogmethodes in twee stappen

Wordt gebruikt om aan te geven welke [twee-staps aanmeldingsmethode](#) moet worden gebruikt bij het [aanmelden](#):

Naam	Waarde
Authenticator	0
E-mail	1
YubiKey	3

Note

FIDO2 en Duo worden niet ondersteund door de CLI.

Soorten items

Wordt gebruikt met het `create` commando om een [vault item type](#) te specificeren:

Naam	Waarde
Inloggen	1
Beveiligde notitie	2
Kaart	3
Identiteit	4

URI-aanmeldingsmatchtypes

Wordt gebruikt met de opdracht **Aanmaken** en **Bewerken** om het detectiegedrag voor URI-overeenkomsten op te geven voor een aanmeldingsitem:

Naam	Waarde
Domein	0
Gastheer	1
Begint met	2
Exact	3
Reguliere expressie	4
Nooit	5

Veldtypen

Wordt gebruikt met de commando's **Maken** en **Bewerken** om aangepaste velden te configureren:

Naam	Waarde
Tekst	0
Verborgen	1
Booleaans	2

Typen organisatiegebruikers

Geeft [het type gebruiker](#) aan:

Naam	Waarde
Eigenaar	0
Admin	1
Gebruiker	2
Manager	3
Aangepast	4

Gebruikersstatussen organisatie

Geeft de status van een gebruiker [binnen de organisatie](#) aan:

Naam	Waarde
Uitgenodigd	0

Naam	Waarde
Geaccepteerd	1
Bevestigd	2
Ingetrokken	-1