

PASSWORD MANAGER > DEVELOPER TOOLS

Lösenordshanteraren CLI

View in the help center:
<https://bitwarden.com/help/cli/>

Lösenordshanteraren CLI

Bitwardens kommandoradsgränssnitt (CLI) är ett kraftfullt, fullt utrustat verktyg för att komma åt och hantera ditt valv. De flesta funktioner som du hittar i andra Bitwarden-klientapplikationer (skrivbord, webbläsartillägg, etc.) är tillgängliga från 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

Bitwarden CLI är självdokumenterat. Lär dig mer om de tillgängliga kommandona från kommandoraden med:

Bash

```
bw --help
```

Eller skicka `--help` som ett alternativ på valfritt bw-kommando för att se tillgängliga alternativ och exempel:

Bash`bw list --help``bw move --help`

Den mesta informationen du behöver kan nås med hjälp av `--help`, men den här artikeln replikerar all information och går in på djupare om vissa ämnen.

Ladda ner och installera

CLI kan användas plattformsoberoende på Windows-, macOS- och Linux-distributioner. För att ladda ner och installera Bitwarden CLI:

Note

For arm64 devices, install the CLI using `npm`.

⇒Native Executable

Natively packaged versions of the CLI are available for each platform and have no dependencies. Download using one of these links:

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

Note that, when using the downloaded native executable, you'll need to add the executable to your PATH or else run commands from the directory the file is downloaded to.

Tip

In Linux and UNIX systems, you might get a `Permission denied` message. If you do, grant permission by running:

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

For each bundle of the Password Manager CLI available on GitHub, there is an OSS (e.g. `bw-oss-windows-2024.12.0.zip`) and non-OSS build (e.g. `bw-windows-2024.12.0.zip`). The non-OSS version is the default package distributed on distribution platforms and includes features under a non-OSS license, such as `device approval` commands, that the OSS version lacks.

Tip

The Bitwarden Password Manager CLI build pipeline creates SHA-256 checksum files that are available on GitHub. [Learn how to validate checksums for the CLI.](#)

⇒NPM

If you have Node.js installed on your system, you can install the CLI using NPM. Installing with NPM is the simplest way to keep your installation up-to-date and should be the **preferred method for those already comfortable with NPM**:

Bash

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

View the package on [npmjs.org](https://www.npmjs.org).

📘 Note

Installing the Bitwarden CLI on Linux systems using **npm** may require the build-essential dependency (or distribution equivalent) to be installed first. For example:

Plain Text

```
apt install build-essential
```

⇒Chocolatey

To install with Chocolatey:

Bash

```
choco install bitwarden-cli
```

View the package on community.chocolatey.org.

⇒Snap

To install with snap:

Bash

```
sudo snap install bw
```

View the package on snapcraft.io.

Logga in

Innan du loggar in, se till att din CLI är ansluten till rätt server (till exempel [EU-moln](#) eller självvärd) med hjälp av kommandot `config` ([läs mer](#)). Det finns tre metoder för att logga in på Bitwarden CLI med hjälp av `login`-kommandot, som var och en är lämpad för olika situationer. Granska följande alternativ för att avgöra vilken metod du ska använda:

- Använd e-post och huvudlösenord
- Använder en API-nyckel
- Använder SSO

Oavsett vilket alternativ du använder, se alltid till att använda kommandona **bw lock** eller **bw logout** när du är klar.

💡 Tip

No matter which option you use, a master password will be required to **unlock** the client in order to access data with a [session key](#). The [email and master password](#) option will authenticate your identity and generate a session key simultaneously, however the [API key](#) or [SSO](#) will require you subsequent use of the **unlock** command to generate a session key if you will be working with data directly.

Users who don't have master passwords, for example as a result of [joining an organization using trusted devices](#), will not be able to access data using the CLI. There are, however, a few commands that do not require decrypted data and therefore can be used without a master password, including **config**, **encode**, **generate**, **update**, and **status**.

Använder e-post och lösenord

Inloggning med e-post och lösenord **rekommenderas för interaktiva sessioner**. För att logga in med e-post och lösenord:

Bash

```
bw login
```

Detta kommer att initiera en uppmaning om din **e-postadress**, **huvudlösenord** och (om aktiverat) vid **tvåstegsinloggningskod**. CLI stöder för närvarande tvåstegsinloggning via [autentisering](#), [e-post](#) eller [Yubikey](#).

Du kan sätta ihop dessa faktorer till ett enda kommando som i följande exempel, men detta rekommenderas inte av säkerhetsskäl:

Bash

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

Se [Enums](#) för tvåstegsinloggning **<method>** värden.

💡 Tip

Getting prompted for additional authentication or getting a **Your authentication request appears to be coming from a bot**. error? Use your API Key **client_secret** to answer the authentication challenge. [Learn more](#).

Använder en API-nyckel

Att logga in med den [personliga API-nyckeln](#) rekommenderas för automatiserade arbetsflöden, för att ge åtkomst till en extern applikation eller om ditt konto använder en 2FA-metod som inte stöds av CLI (FIDO2 eller Duo). För att logga in med API-nyckeln:

Bash

```
bw login --apikey
```

Detta kommer att initiera en prompt för ditt personliga `client_id` och `client_secret`. När din session har autentiserats med dessa värden kan du använda kommandot `upplåsning`. [Läs mer](#).



Tip

If your organization [requires SSO](#), you can still use `--apikey` to log in to the CLI.

Använder API-nyckelmiljövariabler

I scenarier där automatiserat arbete utförs med Bitwarden CLI kan du spara miljövariabler för att förhindra behovet av manuell intervention vid autentisering.

Miljövariabelnamn

BW_CLIENTID

BW_CLIENTSECRET

Obligatoriskt värde

`client_id`

`client_secret`

Använder SSO

Inloggning med [SSO](#) rekommenderas om en organisation kräver SSO-autentisering. För att logga in med SSO:

Bash

```
bw login --sso
```

Detta kommer att initiera [SSO-autentiseringsflödet](#) i din webbläsare. När din session har autentiserats kan du använda `upplåsningskommandot`. [Läs mer](#).



Tip

If your organization [requires SSO](#), you may alternatively use `--apikey` to log in to the CLI.

Logga in på flera konton

Precis som att använda [kontoväxling](#) på andra Bitwarden-appar, har CLI möjligheten att logga in på flera konton samtidigt med hjälp av miljövariabeln `BITWARDENCLI_APPDATA_DIR` som pekar på platsen för en bw-konfigurationsfil, vanligtvis kallad `data.json`. Du kan till

exempel ställa in alias i en **.bashrc-profil** för två separata konfigurationer:

```
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 $@"
```

Med det här exemplet kan du sedan använda inloggning till två konton genom att köra första **källkoden** **/path/to/.bashrc**, följt av **bw-personal login** och **bw-work login**.

Låsa upp

Om du använder en **API-nyckel** eller **SSO** för att logga in måste du följa upp **inloggningskommandot** med en explicit **bw-upplåsning** om du kommer att arbeta med valvdata direkt.

Upplåsning av ditt valv genererar en **sessionsnyckel** som fungerar som en dekrypteringsnyckel som används för att interagera med data i ditt valv. **Sessionsnyckeln måste användas** för att utföra alla kommandon som rör valvdata (till exempel **lista**, **hämta**, **redigera**). Sessionsnycklar är giltiga tills de ogiltigförklaras med **bw lock** eller **bw utloggning**, men de kommer inte att kvarstå om du öppnar ett nytt terminalfönster. Generera en ny sessionsnyckel när som helst med:

```
Bash

bw unlock
```

När du är klar, avsluta alltid din session med **bw lock-kommandot**.

Lås upp alternativ

Du kan använda **--passwordenv<passwordenv>** eller **--lösenordsfil<passwordfile>** alternativ med **bw unlock** för att hämta ditt huvudlösenord istället för att ange det manuellt, till exempel:

1. Följande kommer att leta efter en miljövariabel **BW_PASSWORD**. Om **BW_PASSWORD** inte är tomt och har korrekta värden kommer CLI:n att låsa upp och returnera en sessionsnyckel:

```
Bash

bw unlock --passwordenv BW_PASSWORD
```

2. Följande kommer att leta efter filen **~Users/Me/Documents/mp.txt** (som måste ha ditt huvudlösenord som första rad). Om filen inte är tom och har ett korrekt värde kommer CLI:n att låsa upp och returnera en sessionsnyckel:

```
Bash

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

⚠ Warning

If you use the `--passwordfile` option, protect your password file by locking access down to only the user who needs to run `bw unlock` and only providing read access to that user.

Använda en sessionsnyckel

När du låser upp ditt valv med hjälp av `bw-inloggning` med e-post och lösenord eller `bw unlock`, returnerar CLI både ett `export BW_SESSION` (Bash) och `env:BW_SESSION` (PowerShell) kommando, inklusive din sessionsnyckel. Kopiera och klistra in den relevanta posten för att spara den önskade miljövariabeln.

Med miljövariabeln `BW_SESSION` uppsättning, kommer `bw`-kommandon att referera till den variabeln och kan köras rent, till exempel:

Bash

```
export BW_SESSION="5PBYGU+5yt3RHcCjoeJKx/wByU34vokGRZjXpSH7Ylo8w=="
```

```
bw list items
```

Alternativt, om du inte ställer in miljövariabeln, kan du skicka sessionsnyckeln som ett alternativ med varje `bw`-kommando:

Bash

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

💡 Tip

When you're finished, always end your session using the `bw lock` or `bw logout` commands. This will invalidate the active session key.

Kärnkommandon

skapa

`Skapa` kommandot skapar ett nytt objekt (`objekt`, `bilaga` och mer) i ditt valv:

Bash

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

`Skapa` kommandot tar kodad JSON. Ett typiskt arbetsflöde för att skapa ett objekt kan se ut ungefär så här:

1. Använd kommandot `get template` (se `get core-kommandon` för detaljer) för att mata ut lämplig JSON-mall för objekttypen.

2. Använd en [kommandorads JSON-processor](#) som `jq` för att manipulera den utmatade mallen efter behov.
3. Använd kodningskommandot (se [detaljer](#)) för att koda den manipulerade JSON.
4. Använd kommandot `create` för att skapa ett objekt från den kodade JSON.

Till exempel:

Bash

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

eller

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
```

Efter lyckat skapande kommer det nyskapade objektet att returneras som JSON.

skapa andra objekttyper

Skapa kommandot skapar som standard ett inloggningsobjekt, men du kan använda en [kommandorads JSON-processor](#) som `jq` för att ändra ett `.type=-`attribut för att skapa andra [objekttyper](#):

Namn	Värde
Inloggning	<code>.type=1</code>
Säker anteckning	<code>.type=2</code>
Kort	<code>.type=3</code>
Identitet	<code>.type=4</code>

Till exempel kommer följande kommando att skapa en säker anteckning:

Bash

```
bw get template item | jq '.type = 2 | .secureNote.type = 0 | .notes = "Contents of my Secure Note." | .name = "My Secure Note"' | bw encode | bw create item
```

Note

Notice in the above example that Secure Notes require a sub-template (`.secureNote.type`). You can view item type sub-templates using `bw get template` (see [here](#) for details).

skapa bilaga

Kommandot `create attachment` bifogar en fil till **ett befintligt objekt**.

Till skillnad från **andra skapande operationer** behöver du inte använda en JSON-processor **eller koda** för att skapa en bilaga. Använd istället **alternativet `--file`** för att ange filen som ska bifogas och **alternativet `--itemid`** för att ange objektet att bifoga det till. Till exempel:

Bash

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

Tip

If you don't know the exact `itemid` you want to use, use `bw get item <search-term>` to return the item (see [details](#)), including its `id`.

få

Kommandot `get` hämtar ett enda objekt (**objekt, användarnamn, lösenord** och mer) från ditt valv:

Bash

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

Kommandot `get` tar ett **objekt-id** **eller** en sträng för sitt argument. Om du använder en sträng (till exempel något annat än ett exakt id), kommer `get` att söka i dina valvobjekt efter ett med ett värde som matchar. Till exempel skulle följande kommando returnera ett Github-lösenord:

Bash

```
bw get password Github
```

Note

The **get** command can **only return one result**, so you should use specific search terms. If multiple results are found, the CLI will return an error.

få anknytning

Kommandot **get attachment** laddar ner en filbilaga:

Bash

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

Kommandot **get attachment** tar ett filnamn och exakt id. Som standard kommer **get attachment** att ladda ner bilagan till den aktuella arbetskatalogen. Du kan använda **alternativet --output** för att ange en annan utdatakatalog, till exempel:

Bash

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

Note

When using **--output**, the path **must** end a forward slash (/) to specify a directory or a filename (/Users/myaccount/Pictures/photo.png).

få anteckningar

Kommandot **get notes** hämtar anteckningen för alla valvobjekt:

Bash

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

Kommandot **get notes** tar ett exakt **objekt-id** eller sträng. Om du använder en sträng (till exempel något annat än ett exakt id), kommer **get notes** att söka i dina valvobjekt efter ett med ett värde som matchar. Till exempel skulle följande kommando returnera en Github-anteckning:

Bash

```
bw get notes Github
```

få mall

Kommandot `get template` returnerar den förväntade JSON-formateringen för ett objekt (`item`, `item.field`, `item.login` och mer):

Bash

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

Även om du kan använda `get mall för` att mata ut formatet till din skärm, är det vanligaste användningsfallet att överföra utdata till en `bw create`-operation, med hjälp av en kommandorads JSON-processor som `jq` och `bw encode` för att manipulera värdena som hämtas från mallen, till exempel:

Bash

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

Note

Any `item.xxx` template should be used as a sub-object to an `item` template, for example:

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
```

få fingeravtryck

Hämta `fingeravtrycksfrasen` för den nuvarande inloggade användaren. Du kan ange `användar-ID` direkt eller använda genvägen `mi g`:

Plain Text

```
bw get fingerprint <userId>
```

Plain Text

```
bw get fingerprint me
```

redigera

Redigera kommandot redigerar ett objekt (`objekt`, `objektsamlingar`, etc.) i ditt valv:

Bash

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

Redigera kommandot tar ett **exakt id** (objektet som ska redigeras) och kodad JSON (redigeringar ska göras). Ett typiskt arbetsflöde kan se ut ungefär så här:

1. Använd kommandot **get** (se [detaljer](#)) för att mata ut objektet som ska redigeras.
2. Använd [en kommandorads JSON-processor](#) som **jq** för att manipulera det utmatade objektet efter behov.
3. Använd kodningskommandot (se [detaljer](#)) för att koda den manipulerade JSON.
4. Använd kommandot **edit** (inklusive **objekt-id**) för att redigera objektet.

Till exempel, för att redigera lösenordet för ett inloggningsobjekt:

Bash

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

Eller för att redigera samlingen/samlingarna som ett objekt finns i:

Bash

```
echo '["5c926f4f-de9c-449b-8d5f-aec1011c48f6"]' | bw encode | bw edit item-collections 28399a57-73a  
0-45a3-80f8-aec1011c48f6 --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5
```

Eller för att redigera en samling:

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
```

Redigera kommandot kommer att utföra en ersättningsoperation på objektet. När det är klart kommer det uppdaterade objektet att returneras som JSON.

lista

Listkommandot hämtar en rad objekt (**objekt**, **mappar**, **samlingar** och mer) från ditt valv:

Bash

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

Alternativ för listkommandot är **filter som används** för att diktera vad som kommer att returneras, inklusive `--url<url>` , `--folderid<folderid>` , `--collectionid<collectionid>` , `--organisationid<organizationid>` och `--sopor`. Alla filter accepterar null eller inte null. Att kombinera flera filter i ett kommando kommer att utföra en ELLER-operation, till exempel:

Bash

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

Detta kommando returnerar objekt som inte finns i en mapp eller samling.

Dessutom kan du **söka efter** specifika objekt med `--search<search-term>` . Genom att kombinera filter och sökning i ett kommando utförs en OCH-operation, till exempel:

Bash

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

Detta kommando kommer att söka efter objekt med strängen `github` i den angivna mappen.

radera

Kommandot `delete` tar bort ett objekt från ditt valv. `delete` tar bara **ett exakt id** för sitt argument.

Bash

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

Som standard kommer **radering** att skicka ett objekt till [papperskorgen](#), där det finns kvar i 30 dagar. Du kan ta bort ett objekt permanent med alternativet `-p`, `--permanent`.

Bash

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

För att ta bort en **organisationssamling** måste du också ange `--organizationid<organizationid>` . Se [Organisations-ID:n](#).

⚠ Warning

While items that are deleted using **delete** can be recovered using the **restore** command for up to 30 days (see [details](#)), items that are deleted using **delete --permanent** are **completely removed and irrecoverable**.

återställa

Kommandot **återställ** återställer ett borttaget objekt från papperskorgen. **restore** tar bara **ett exakt id** **för** sitt argument.

Bash

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

Till exempel:

Bash

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

skicka

Kommandot **skicka** skapar ett **Bitwarden Send**-objekt för tillfällig delning. Det här avsnittet beskriver enkla sändningsoperationer, men **skicka** är ett mycket flexibelt verktyg och vi rekommenderar att du hänvisar till den dedikerade artikeln [om Skicka från CLI](#).

Så här skapar du en enkel text Skicka:

Bash

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

För att skapa en enkel fil Skicka:

Bash

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

få

Mottagningskommandot kommer åt ett **Bitwarden Send**-objekt. Så här tar du emot ett sändobjekt:

Bash

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

Organisationers kommandon

Organisations-ID

För att komma åt en organisation från CLI krävs kunskap om ett ID för din organisation, samt ID:n för [enskilda medlemmar](#) och [samlingar](#).

Hämta denna information direkt från CLI med hjälp [av](#) `bw-listan`, till exempel:

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

You can `bw list` both `collections` and `org-collections`. The `bw list collections` command will list all collections, agnostic of which organization they belong to. `bw list org-collections` will list only collections that belong to the organization specified using `--organizationid`.

flytta

Note

August 2021: The `share` command has been changed to `move`. [Find out more](#).

Flytta-kommandot överför ett valvobjekt till [en organisation](#):

Bash

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

Flytta-kommandot kräver att du [kodar](#) ett samlings-ID och tar [ett exakt](#) id (objektet att dela) och [ett exakt](#) organisations-id (organisationen att dela objektet till). Till exempel:

Bash

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

När det är klart kommer det uppdaterade föremålet att returneras.

bekräfta

Kommandot **bekräfta** bekräftar inbjudna medlemmar till din organisation som har accepterat deras inbjudan:

Bash

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

Kommandot **bekräfta** tar ett exakt medlems-ID och ett exakt organisations-ID, till exempel:

Bash

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

Enhetsgodkännande

Tillåter administratörer och ägare att hantera begäranden om enhetsgodkännande där en användare har begärt administratörsgodkännande.

Note

At this time, bulk device approval is only available for the Bitwarden CLI client downloaded from [Bitwarden.com](https://bitwarden.com).

Warning

In most scenarios, users are able to approve their own login requests, and admin device approval is not necessary. See [Add a trusted device](#). Automatic or bulk approval of admin device approval requests neglect verification steps that administrators can perform in order to ensure a request is legitimate, such as checking the user's reported Fingerprint Phrase.

Bitwarden recommends that significant security controls such as IdP credential standards, IdP MFA, and IdP device registration and trust be reviewed before enabling and using bulk device approval.

Listkommandot visar alla väntande begäranden om enhetsgodkännande för en organisation:

Plain Text

```
bw device-approval list --organizationid <organization_Id>
```

Godkänn kommandot används för att godkänna väntande begäranden om enhetsauktorisering för en organisation:

Plain Text

```
bw device-approval approve --organizationid <organizationId> <requestId>
```

På samma sätt kan **approve-all-kommandot** användas för att godkänna alla aktuella väntande förfrågningar:

Plain Text

```
bw device-approval approve-all --organization <organizationId>
```

Så här **avvisar** du en väntande auktoriseringsbegäran:

Plain Text

```
bw device-approval deny --organizationid <organizationId> <requestId>
```

Så här nekar du **alla** väntande auktoriseringsförfrågningar:

Plain Text

```
bw device-approval deny-all --organizationid <organizationId>
```

Andra kommandon

config

Kommandot config anger inställningar som Bitwarden CLI ska använda:

Bash

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

En primär användning av bw **config** är att ansluta din CLI till en Bitwarden-server som är självvärd:

Bash

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

 Tip

Connect to the Bitwarden [EU server](#) by running the following command:

Bash

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

Pass `bw config server` without a value to read the server you're connected to.

Användare med unika inställningar kan välja att ange URL:en för varje tjänst oberoende. Observera att varje efterföljande användning av `config`-kommandot kommer att skriva över alla tidigare specifikationer, så detta måste köras som ett enda kommando varje gång du gör en ändring:

Bash

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

 Note

The `bw config server --key-connector <url>` command is required if your organization uses [Key Connector](#) and you're using the `--apikey` option to login after having [removed your master password](#).

Contact an organization owner to get the required URL.

synkronisera

Synkroniseringskommandot laddar ner ditt krypterade valv från Bitwarden-servern. Det här kommandot är mest användbart när du har ändrat något i ditt Bitwarden-valv på en annan klientapplikation (till exempel webbvalv, webbläsartillägg, mobilapp) [sedan du loggade in](#) på CLI.

Bash`bw sync`

Du kan skicka **alternativet** `--last` för att endast returnera tidsstämpeln (ISO 8601) för senaste gången en synkronisering utfördes.

💡 Tip

It's important to know that **sync only performs a pull** from the server. Data is automatically pushed to the server any time you make a change to your vault (for example, **create, edit, delete**).

koda

Koda kommandot Base 64 kodar stdin. Detta kommando används vanligtvis i kombination med en **kommandorads JSON-processor** som jq när du utför skapa och redigera operationer, till exempel:

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
```

importera

Importkommandot importerar data från en Bitwarden-export eller **annat lösenordshanteringsprogram** som stöds. Kommandot måste peka på en fil och innehålla följande argument:

Bash`bw import <format> <path>`

Till exempel:

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

 Tip

Bitwarden supports lots of formats for import, too many to list here! Use `bw import --formats` to return the list in your CLI, or [see here](#).

If you are importing an [encrypted json file](#) that you've created with a password, you will be prompted to enter the password before import completes.

importera till ett organisationsvalv

Genom att använda importkommandot med alternativet `--organizationid` kan du importera data till ett organisationsvalv:

Plain Text

```
bw import --organizationid cf14adc3-aca5-4573-890a-f6fa231436d9 bitwardencsv ./from/source.csv
```

exportera

Exportkommandot exporterar valvdata som en `.json-` eller `.csv-`, krypterad json-fil eller `.zip` som innehåller [bilagor](#):

Bash

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

Som standard genererar **exportkommandot** en `.csv` (motsvarande att specificera `--format csv`) till den aktuella arbetskatalogen, men du kan ange:

- `--format json` för att exportera en `.json-fil`
- `--format encrypted_json` för att exportera en krypterad json-fil
 - `--lösenord<password>` för att ange ett lösenord som ska användas för att kryptera **encrypted_json-exporter** istället för ditt [kontos krypteringsnyckel](#)
- `-- formatera zip` för att exportera en `.zip-fil` som innehåller [bilagor](#) (endast individuella valvdata)
- `--produktion<path>` för att exportera till en specifik plats
- `--raw` för att returnera exporten till stdout istället för till en fil

export från ett organisationsvalv

Genom att använda **exportkommandot** med alternativet `--organizationid` kan du exportera ett organisationsvalv:

Bash

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

generera

Kommandot generera genererar ett starkt lösenord eller lösenfras:

Bash

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

Som standard genererar **kommandot** generera ett lösenord med 14 tecken med versaler, gemener och siffror. Detta motsvarar godkänt:

Bash

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

Du kan skapa mer komplexa lösenord med hjälp av de alternativ som är tillgängliga för kommandot, inklusive:

- **--versaler**, **-u** (inkludera versaler)
- **--små** bokstäver, **-l** (inkludera gemener)
- **--nummer**, **-n** (inkludera siffror)
- **--special**, **-s** (inkluderar specialtecken)
- **--längd<length>** (lösenordets längd, min av 5)

generera en lösenfras

Genom att använda **kommandot generera** med alternativet **--passphrase** kan du generera en lösenordsfras istället för ett lösenord:

Bash

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

Som standard kommer **bw generera --lösenfras** att generera en treordslösenordsfras separerad med ett bindestreck (-). Detta motsvarar godkänt:

Bash

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

Du kan generera en komplex lösenfras med hjälp av de alternativ som är tillgängliga för kommandot, inklusive:

- `--ord<words>` (antal ord)
- `--separator<separator>` (separatortecken)
- `--capitalize, -c` (inkludera lösenfrasen i rubriken-versaler)
- `--includeNumber` (Inkludera ett enda numeriskt tecken i din lösenfras)

uppdatera

Uppdateringskommandot kontrollerar om din Bitwarden CLI kör den senaste versionen. `update` uppdaterar **inte automatiskt CLI åt dig**.

Bash

```
bw update
```

Om en ny version upptäcks måste du ladda ner den nya versionen av CLI:n med den utskrivna URL:en för den körbara filen, eller använda de tillgängliga verktygen för pakethanteraren som du använde för [att ladda ner CLI:n](#) (till exempel `npm install -g @bitwarden/cli`).

status

Statuskommandot returnerar statusinformation om Bitwarden CLI, [inklusive konfigurerad](#) server-URL, tidsstämpel för den senaste synkroniseringen (ISO 8601), användarens e-post och ID och valvstatus.

Bash

```
bw status
```

Status returnerar information som ett JSON-objekt, till exempel:

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 vara något av följande:

- **"upplåst"**, vilket indikerar att du är inloggad och ditt valv är upplåst (en **BW_SESSION**-nyckelmiljövariabel sparas med en **aktiv sessionsnyckel**)
- **"låst"**, vilket indikerar att du är inloggad men ditt valv är låst (**ingen BW_SESSION**-nyckelmiljövariabel sparas med en **aktiv sessionsnyckel**)
- **"oautenticerad"**, vilket indikerar att du inte är inloggad

**Tip**

When **"status": "unauthenticated"**, **lastSync**, **userEmail**, and **userID** will always return **null**.

tjäna

Serva-kommandot startar en lokal expresswebbserver som kan användas för att utföra alla åtgärder som är tillgängliga från CLI i form av RESTful API-anrop från ett HTTP-gränssnitt.

Bash

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

Som standard startar **serve** webbservern vid port 8087 men du kan ange en alternativ port med alternativet **--port**.

Som standard kommer **servern** att binda din API-webbserver till **localhost** men du kan ange ett alternativt värddamn med alternativet **--hostname**. API-förfrågningar kan endast göras från det bundna värddamnet.

Som standard blockerar **servern** alla förfrågningar med en **Origin-rubrik**. Du kan kringgå detta skydd med alternativet **--disable-origin-protection**, men **detta rekommenderas** inte.

⚠ Warning

You can specify `--hostname all` for no hostname binding, however this will allow any machine on the network to make API requests.

Se [API-specifikationen](#) för att få hjälp att ringa med `serve`.

Felsök

Felsökningsmiljövariabeln kan läggas till för ytterligare felsökningsinformation.

Plain Text

```
export BITWARDENCLI_DEBUG=true
```

Bilagor

Globala alternativ

Följande alternativ är tillgängliga globalt:

Alternativ	Beskrivning
<code>--ganska</code>	Formatera utdata. JSON är flikar med två mellanslag.
<code>--rå</code>	Returnera rå utdata istället för ett beskrivande meddelande.
<code>--svar</code>	Returnera en JSON-formaterad version av svarsutdata.
<code>--tyst</code>	Lämna inte tillbaka något till stdout. Du kan till exempel använda det här alternativet när du överför ett autentiseringsvärde till en fil eller ett program.
<code>-- ingen interaktion</code>	Fråga inte efter interaktiv användarinmatning.
<code>--session<session></code>	Passera sessionsnyckel istället för att läsa från en miljövariabel.

Alternativ	Beskrivning
<code>-v, --version</code>	Skriv ut Bitwardens CLI-versionsnummer.
<code>-h, --hjälp</code>	Visa hjälptext för kommandot.

Komplettering av ZSH-skäl

Bitwarden CLI inkluderar stöd för ZSH-skalkomplettering. Använd någon av följande metoder för att konfigurera skalkomplettering:

1. **Vanilla ZSH:** Lägg till följande rad i din `.zshrc`-fil:

Bash

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

2. **Vanilla (leverantörskompletteringar):** Kör följande kommando:

Bash

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

3. **zinit:** Kör följande kommandon:

Bash

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

Använder självsignerade certifikat

Om din självvärdade Bitwarden-server visar ett självsignerat TLS-certifikat, ange miljövariabeln Node.js `NODE_EXTRA_CA_CERTS`:

🍏 Våldsamt slag

Bash

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

Bash

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

Enums

Följande tabeller räknar upp värden som krävs i dokumenterade scenarier:

Inloggningsmetoder i två steg

Används för att ange vilken tvåstegsinloggningsmetod [som ska användas när du loggar in](#):

Namn	Värde
Autentisering	0
E-post	1
YubiKey	3

Note

FIDO2 and Duo are not supported by the CLI.

Objekttyper

Används med kommandot **create** för att ange en typ av valvobjekt:

Namn	Värde
Inloggning	1
Säker anteckning	2

Namn	Värde
Kort	3
Identitet	4

Inloggnings-URI-matchningstyper

Används med kommandot **skapa** och **redigera** för att ange **URI-matchningsdetekteringsbeteende** för ett inloggningsobjekt:

Namn	Värde
Domän	0
Värd	1
Börjar med	2
Exakt	3
Reguljärt uttryck	4
Aldrig	5

Fälttyper

Används med kommandona **skapa** och **redigera** för att konfigurera **anpassade fält**:

Namn	Värde
Text	0
Dold	1
Boolean	2

Organisationsanvändartyper

Indikerar en användares [typ](#):

Namn	Värde
Ägare	0
Administration	1
Användare	2
Chef	3
Beställnings	4

Organisationens användarstatus

Indikerar en användares [status](#) inom organisationen:

Namn	Värde
Inbjudna	0
Accepterad	1
Bekräftad	2
Återkallad	-1