# Bitwarden CLI Vault from the shell

**START HERE**

`bw login`
Sign in (asks for password)

`bw get password <name>`
Print one password

`bw unlock --raw`
Print a session key

`bw generate -uln --length 20`
Make a password

`export BW_SESSION="<key>"`
Use the key in this shell

`bw sync`
Pull the latest vault

`bw list items --search <text>`
Find entries

`bw lock`
End the session

**Logged out** no account → `bw login` → **Locked** data encrypted → `bw unlock` → **Unlocked** BW_SESSION set → `bw lock` → **Locked** key destroyed

## CONCEPTS

Logging in is not unlocking: the vault stays encrypted until you have a session key in BW_SESSION or --session.

Session keys and master passwords on the command line end up in shell history and process lists; prefer prompts, --passwordenv or --passwordfile.

Commands that need no unlock: config, encode, generate, status, update.

## SIGN IN

`bw login`
Sign in with email and password

`bw login --apikey`
Sign in with an API key
*Prompts for client_id and client_secret; BW_CLIENTID and BW_CLIENTSECRET skip the prompt. Run bw unlock afterwards.*

`bw login --sso`
Sign in through your SSO provider

`bw unlock`
Unlock the vault

`bw unlock --raw`
Print only the session key

`bw unlock --passwordenv <VAR>`
Read master password from an env var

`bw unlock --passwordfile <file>`
Read master password from a file
*Takes the first line; keep the file readable only by you.*

`bw lock`
Lock the vault, destroy the key

`bw logout`
Sign out of the account

`bw status`
Show login, lock and sync state

`bw config server <url>`
Point at a self-hosted server

`bw --session <key> <command>`
Pass the session key per command

## FIND AND READ

`bw list items`
List all vault items as JSON

`bw list items --search <text>`
Search items

`bw list items --folderid <id>`
Items in one folder

`bw list items --url <url>`
Items matching a URL

`bw list folders`
List folders

`bw get item <name|id>`
Show one full item
*A search string works instead of an id, but several matches give an error; use the id.*

`bw get password <name|id>`
Print the password

`bw get username <name|id>`
Print the username

`bw get totp <name|id>`
Print the current 2FA code

`bw get notes <name|id>`
Print the notes

`bw get attachment <file> --itemid <id>`
Download an attachment
*With --output, a directory path must end in a slash.*

## CREATE AND CHANGE

`bw get template item`
Print an item JSON template

`bw encode`
Base64-encode JSON from stdin

`bw create item <encodedJson>`
Create an item

`bw create folder <encodedJson>`
Create a folder

`bw edit item <id> <encodedJson>`
Replace an item
*Replaces the whole object, so edit the output of bw get item.*

`bw edit folder <id> <encodedJson>`
Rename a folder

`bw delete item <id>`
Move an item to the trash

## CREATE AND CHANGE (cont.)

`bw delete item <id> --permanent`
Delete an item for good

`bw restore item <id>`
Restore from the trash

`bw create attachment --file <path> --itemid <id>`
Attach a file to an item

`bw get template folder`
Print the folder template

## GENERATE AND SYNC

`bw generate`
Make a password (14 chars default)

`bw generate -uln --length 24`
Letters and digits, 24 long

`bw generate -ulns --length 24`
Add special characters

`bw generate --passphrase --words 5`
Make a passphrase

`bw generate -p --separator - -c`
Title-case words joined by -

`bw generate --ambiguous`
Avoid look-alike characters

`bw sync`
Pull the latest vault data

`bw sync --force`
Force a full sync

`bw sync --last`
Show the last sync time

## SEND

`bw send -n "<name>" -d <days> --hidden "<text>"`
Send text, hidden by default

`bw send -n "<name>" -d <days> -f <path>`
Send a file

`bw send list`
List your Sends

`bw send delete <id>`
Delete a Send

`bw send receive <url>`
Open a Send by URL

`bw send receive --password <pw> <url>`
Open a protected Send

## SEND (cont.)

`bw send --help`
Show all Send subcommands
*Documents remove-password, which the web page does not cover.*

## BACKUP AND OPTIONS

`bw export`
Export the vault as CSV

`bw export --format json --output <dir>/`
Export as JSON to a folder

`bw export --format encrypted_json --password`
Password-encrypted export
*Without --password, the export is tied to your account key.*

`bw import --formats`
List supported import formats

`bw import <format> <path>`
Import a file into the vault

`bw --pretty`
Indent JSON output

`bw --raw`
Return plain values, not messages

`bw --nointeraction`
Never prompt

`bw --help`
Show all commands

`bw <command> --help`
Help for one command

## RECIPES

**Unlock in the current shell**

```
export BW_SESSION="$(bw unlock --raw)"
bw sync
bw get password <name>
```

**Change an item's password**

```
bw get item <id> \
  | jq '.login.password="<new>"' \
  | bw encode | bw edit item <id>
```

**Create a folder**

```
bw get template folder \
  | jq '.name="<name>"' \
  | bw encode | bw create folder
```

**Backup, then clear the file**

```
bw sync
bw export --format encrypted_json \
  --password --output <dir>/
```