Bittensor Documentation

Do anything on the Bittensor chain with the Bittensor SDK and CLI.

Copy Markdown View as Markdown

Bittensor is an open network where independent **subnets** produce digital commodities — compute, inference, storage, prediction — and the chain pays participants in its token, **TAO**, in proportion to the

Archived copy · cryp2nova.com

value they contribute. **Miners** produce the commodity, **validators** score the miners, **subnet creators** define the incentive mechanism, and **stakers** back validators with TAO.

These docs cover the `bittensor` Python SDK and CLI: one install gives you a library (`import bittensor`) and a command line (`btcli`) that together can perform every user-facing operation on the chain.

Coming from the v10 SDK or the standalone `bittensor-cli`? The API changed substantially in v11 — see Migrating from v9/v10 for exact old-to-new mappings.

Where to go

Quickstart

Install, connect, first query, first transaction — in five minutes.

The SDK

The five core primitives — Subtensor, Wallet, Balance, intents, results — on one page.

For agents

How an agent discovers, previews, and safely executes any chain operation.

Transactions

Every state-changing operation, one page each, generated from the SDK.

Queries

Every chain read, one page each, generated from the SDK.

Errors

Every failure carries a machine-readable code and a remediation hint.

Concepts

The network, wallets, money, the client, and the transaction model.

Migrating from v9/v10

Exact mappings from the legacy SDK and btcli to the unified v11 package.

Install

Requires Python 3.10–3.14. Using uv:

```
uv venv && source .venv/bin/activate
uv pip install bittensor
```

Archived copy · cryp2nova.com

This installs both the `btcli` command and the `bittensor` Python package.

Machine-readable everything

If you are an agent (or building one), you never need to scrape these pages. See For agents → Searching these docs for the curl/grep recipes.

- `/llms.txt` — curated index; every docs link in it is already raw markdown (`content.md`).
- `/llms-full.txt` — entire docs corpus in one file (for `rg`, not for stuffing into context).
- Every page as raw markdown: `/llms.mdx/docs/<slug>/content.md` (also linked from each page's copy-markdown control).
- `/catalog/intents.json` — every transaction with its JSON schema, signer, `markdown_url`, and summary.
- `/catalog/reads.json` — every query with parameters, `markdown_url`, and docs.
- `/catalog/errors.json` — semantic codes under `.codes`, chain names under `.chain_errors`; each has remediation + `markdown_url`.
- Chain Rust source: `/code/search.json?q=…` to search, `/code/index.json` to list, `/code/raw/<repo-path>` to fetch.
- The same catalogs, live from the tool itself: `btcli tools` on the CLI, `bt.intents.list_tools()` in Python.

On this page

- Where to go
- Install
- Machine-readable everything

CRYP2NOVA

Archived copy · cryp2nova.com