

Contents

- [Resources](#)
- [Wallets](#)
- [Client Libraries](#)
 - [Go](#)
 - [JavaScript](#)
 - [Python](#)
 - [Rust](#)
- [Block Explorers](#)
- [Visual Block Explorer](#)
- [Chain Registry](#)
- [Validators](#)
- [Modules](#)
 - [External Modules](#)
- [Monitoring](#)
- [Frameworks](#)
- [Virtual Machines](#)
- [IBC](#)
- [Bridges](#)
- [Tools](#)
- [Ecosystem](#)
- [Blogs](#)
 - [Articles](#)
- [Related](#)

Resources

- [Cosmos SDK](#)
- [IBC Go](#)
- [Tendermint](#)
- [CosmWasm](#)
- [CosmJS](#)
- [Cosmos Hub \(Gaia\)](#)
- [Cosmos Developer Portal](#)
- [Docs - Cosmos SDK](#)
- [Docs - IBC](#)
- [Docs - Tendermint](#)
- [Docs - Cosmos Hub \(Gaia\)](#)
- [Docs - CosmWasm](#)
- [Docs - Cosmology](#)

Wallets

The most up-to-date list of wallets supporting cosmos chains is <https://cosmos.network/ecosystem/wallets>.

Client Libraries

Go

- [Ignite CLI](#) - All-in-one platform to build, launch, and maintain any crypto application on a sovereign and secured blockchain. Quickly bootstraps a new Cosmos SDK blockchain with UI and support to create new and work conveniently with existing Cosmos SDK modules.

JavaScript

- [cosmos/cosmjs](#) - **ICF sponsored Cosmos JavaScript library**.
- [telescope](#) - Typescript library generator built on top of CosmJS.
- [chainapsis/cosmosjs](#) - Chainapsis Signing & API Library.
- [cosmos-client/cosmos-client-ts](#) - JavaScript / TypeScript client for Cosmos SDK blockchains.
- [cosmos/sig](#) - Cosmos Signing Library.
- [cosmology-tech/chain-registry](#) - A npm package for the official Cosmos Chain Registry.

Python

- [cosmospy](#) - Tools for Cosmos wallet management and offline transaction signing.
- [cosmpy](#) - A Python client library for interacting with blockchains based on the Cosmos-SDK.

Rust

- [CosmWasm/cosmwasm](#) - WebAssembly Smart Contracts for the Cosmos SDK.
- [iglusioninc/stdtx](#) - A collection of open source Rust crates from iglusion.
- [peggyiv/ocular](#) - A client library for Cosmos SDK chains focusing on pleasant UX.

Block Explorers

- [ATOMScan](#)
- [Big Dipper](#) - [Source](#)
- [IOBScan](#)
- [Mintscan](#)
 - [Mintscan for Cosmos Hub Testnet](#)
- [NG Explorers](#)
- [Ping.pub](#) - [Source](#)
- [Stake ID](#)

Visual Block Explorer

View Inter-Blockchain Communication (IBC) transfer activity. The map traces IBC transactions between different blockchains (called zones in the Cosmos Hub) to display accurate aggregate information about the pulse of the entire Cosmos ecosystem.

- [Map of Zones](#) - [Source](#)
- [Mintscan Hub](#)

Chain Registry

A registry containing standardized metadata from most Cosmos chains.

- [cosmos/chain-registry](#)
- [Cosmos Directory](#)
- [cosmology-tech/chain-registry](#) - A npm package for the official Cosmos Chain Registry.

Validators

Popular block explorers provide a list of active validators. The easiest entry point to view validator profiles is from a block explorer. For example, [Mintscan](#), [ATOMScan](#), and [BigDipper](#). DYOR when choosing a validator. Consider delegating your tokens to validators outside of the top 20 to increase the decentralization of the network.

Modules

The best place to find an accurate list of the Cosmos SDK modules is the project repository. Find modules in <https://github.com/cosmos/cosmos-sdk/tree/main/x>. For a list of production-grade modules, see [List of Modules](#). Some modules are important but may still lack official documentation, like the [genutil](#) module that is essential to launch a chain. Please consider contributing documentation to the Cosmos SDK to help fill these gaps.

To share a module that you want the community to know about and use, add it here. You can also create a pull request (PR) to add the module to the official project.

External Modules

- [Ethermint - EVM](#)
- [Cosmos - Gravity Bridge](#)
- [Cosmos/Modules - Faucet](#)
- [Kava - Auction](#)
- [Kava - CDP](#)
- [Kava - Pricefeed](#)
- [Kava - Validator Vesting](#)
- [Tendermint - Liquidity](#)

Monitoring

- [PANIC Monitoring and Alerting For Blockchains](#) - An open source monitoring and alerting solution for Cosmos SDK, Substrate, and Chainlink-based nodes.
- [Prometheus Exporter](#) - An integrated Prometheus exporter for the Cosmos SDK.
- [Cosmos Chains Dashboard](#) - A Grafana dashboard to monitor Cosmos SDK and Tendermint-based blockchain nodes.
- [ibc.cosmos.network](#) - IBC-Go official documentation.
- [missed-blocks-checker](#) - Monitor missed blocks for Cosmos-chain validators with support for notifications to Telegram and Slack.
- [Nodes Checker](#) - Check your nodes status online, receive instant notification if something is wrong with your validator node.
- [Cosmon](#) - Observability tool for Cosmos and other Tendermint applications.

Frameworks

- [Cosmos SDK](#) - A Framework for Building High Value Public Blockchains in Go.
- [Orga](#) - ABCI framework for state machine transitions in Rust.
- [CosmosSwift](#) - Build blockchain applications in Swift on top of the Tendermint consensus.
- [ABCI-RS](#) - Rust crate for creating ABCI applications.
- [CosmRS](#) - Framework for building Cosmos blockchain applications in Rust.

Virtual Machines

Modules or frameworks for virtual machines that run in the Cosmos SDK

- [Agoric SDK](#) - Agoric JavaScript Smart Contract Platform.
- [CosmWasm](#) - WASM Virtual Machine & Rust Smart Contracts.
- [Ethermint](#) - Ethereum Virtual Machine.

IBC

- [IBCprotocol.org](#) - IBC Protocol website.
- [Interchain Standards](#) - Interchain Standards (ICS) for the Cosmos network & interchain ecosystem.
- [ibc.cosmos.network](#) - IBC-Go official documentation.
- [cosmos/ibc-go](#) - Inter-Blockchain Communication protocol (IBC) implementation in Go.
- [informalsystems/ibc-rs](#) - IBC data structures and handlers, and a relay for Cosmos SDK chains written in Rust.
- [cosmos/relayer](#) - An IBC relayer written in Go for `ibc-go`.
- [ibctest](#) - Conformance tests for inter-blockchain communication.

Bridges

- [ctopus-network/substrate-ibc](#) - IBC on Substrate.
- [ChorusOne/quantum-tunnel](#) - Cosmos <-> Substrate Relayer.
- [nomic-io/nomic](#) - Merge-mined Bitcoin sidechain built with Orga and Tendermint.

Tools

CLI

- [iglusioninc/tmkms](#) - Key Management System for Tendermint validators.
- [cosmosvisor](#) - Automates Cosmos SDK application binary upgrades.
- [cosmosvanity](#) - CLI tool for generating cosmos vanity addresses.
- [findaccount](#) - Helps identify if an account exists on multiple cosmos chains with the same address.
- [lens](#) - CLI tool to interact with any cosmos chain supporting the core Cosmos-SDK modules.
- [cosmology](#) - CLI tool for making cryptocurrency trades, joining liquidity pools, and stake rewards on Osmosis.
- [multisig](#) - CLI tool for managing multisig accounts on Cosmos SDK chains.

GUI

- [Cosmos Hub Multisig Manager](#) - Create multisigs and multisig transactions on the Cosmos Hub ([source](#)).

Ecosystem

The most up-to-date list of projects built using the Cosmos SDK is <https://cosmos.network/ecosystem/apps>.

Blogs

Disclaimer: This community-maintained repo does not reflect the views of any official entity.

As the ecosystem grows, so does the content. DYOR and follow the projects you find interesting.

- [What is Cosmos?](#)
- [Cosmos Blog](#)
- [Interchain Foundation Blog](#)

Articles

- [Cosmos Dev Series: Cosmos Blockchain Upgrade](#)
- [Hardening guides](#)

Related

- [Awesome Tendermint consensus](#)
- [Awesome Binance Chain](#)
- [Awesome CosmWasm](#)
- [Awesome Crypto.com Chain](#)
- [Awesome Evmos](#)
- [Awesome Osmosis](#)