[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 0 - Getting Started](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Getting Started](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Blockchain 101](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Blockchain History](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Public and Managed Blockchains](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Consensus in Distributed Networks](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Cryptography](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Self-Assessment Quiz](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Go Introduction - First Steps](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Go Basics](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Go Interfaces](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Control Structures in Go](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Arrays and Slices in Go](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Standard Packages in Go](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Concurrency in Go](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Docker Introduction](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Introduction to the Interchain](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[The Interchain Ecosystem](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Getting ATOM and Staking It](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[A Blockchain App Architecture](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Accounts](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Transactions](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Messages](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Modules](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Protobuf](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Multistore and Keepers](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[BaseApp](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Queries](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Events](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Context](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Testing](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Relaying with IBC](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Interchain Security](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Bridges](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Migrations](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 1 Quiz](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 2 - First Steps](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[First Steps](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Setup Your Work Environment](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Run a Node, API, and CLI](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Ignite CLI](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Store Object](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Create Custom Messages](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Create and Save a Game Properly](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Add a Way to Make a Move](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Emit Game Information](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Record the Game Winner](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 2 Exercise](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[What is IBC?](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[IBC Token Transfer](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[IBC Tooling](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[What is CosmJS?](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Your First CosmJS Actions](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Compose Complex Transactions](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Learn to Integrate Keplr](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Put Your Games in Order](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Auto-Expiring Games](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Let Players Set a Wager](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Handle wager payments](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Integration tests](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Incentivize Players](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Help Find a Correct Move](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Understand IBC Denoms](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Go Relayer](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Hermes Relayer](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[CosmJS Advanced](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Create Custom Objects](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Create Custom Messages](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Get an External GUI](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Backend Script for Game Indexing](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[IBC Deep Dive](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[IBC Application Developer Introduction](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Create a Leaderboard Chain](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Run in Production](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Prepare the Software to Run](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Prepare a Validator and Keys](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Prepare Where the Node Starts](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Prepare and Do Migrations](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Simulate Production in Docker](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Tally Player Info After Production](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Simulate a Migration in Docker](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Final Exam](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[What's Next?](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

[Continue Your Interchain Journey](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html)

Docs Version Switcher

On this page

[Context properties](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#context-properties)

[Golang Context Package](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#golang-context-package)

[Store branching](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#store-branching)

[The pattern of usage](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#the-pattern-of-usage)

[Process](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#process)

[#Copy link](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#context) **Context**



It is recommended to first read the following sections to better understand context:

* [Transactions](https://ida.interchain.io/academy/2-cosmos-concepts/3-transactions.html)
* [Messages](https://ida.interchain.io/academy/2-cosmos-concepts/4-messages.html)
* [Modules](https://ida.interchain.io/academy/2-cosmos-concepts/5-modules.html)
* [BaseApp](https://ida.interchain.io/academy/2-cosmos-concepts/8-base-app.html)
* [Queries](https://ida.interchain.io/academy/2-cosmos-concepts/9-queries.html)
* [Events](https://ida.interchain.io/academy/2-cosmos-concepts/10-events.html)



Context is the setting in which transactions execute, and is the sum of all pertinent information at runtime. Here you will find out what transaction context means in detail and learn more about the important elements that together form the execution context.

Transactions execute in a context. The context includes information about the current state of the application, the block, and the transaction.

Context is represented as data structures that carry information about the current state of the application and are intended to be passed from function to function. Context provides access to branched storage, that is a safe branch of the entire state, as well as useful objects and information, like gasMeter, block height, and consensus parameters.



The Cosmos SDK context is a custom data structure that contains Go's stdlib context as its base. It has many additional types within its definition that are specific to the Cosmos SDK.

Context is integral to transaction processing as it allows modules to easily access their respective store in the multistore and retrieve transactional context such as the block header and gas meter.

[#Copy link](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#context-properties) Context properties

The context has the following properties:

* **Context:** the base type is a Go Context.
* **Multistore:** every application's BaseApp contains a CommitMultiStore, which is provided when a context is created. Calling the KVStore() and TransientStore() methods allows modules to fetch their respective KVStores using their unique StoreKeys.
* **ABCI Header:** the header is an ABCI type. It carries important information about the state of the blockchain, such as block height and the proposer of the current block.
* **Chain ID:** the unique identification number of the blockchain a block pertains to.
* **Transaction bytes:** the []byte representation of a transaction is processed using the context.



Every transaction is processed by various parts of the Cosmos SDK and consensus engine (for example CometBFT) throughout its lifecycle, some of which do not have any understanding of transaction types. Thus, transactions are marshaled into a generic []byte type using some kind of encoding format such as Amino.

* **Logger:** a logger from the Tendermint libraries. [Learn more about logs here (opens new window)↗](https://github.com/tendermint/tendermint/blob/master/libs/log/logger.go). Modules call this method to create their unique module-specific logger.
* **VoteInfo:** a list of the ABCI type VoteInfo, which includes the name of a validator and a boolean indicating whether they have signed the block.
* **Gas meters:** specifically, a gasMeter for the transaction currently being processed, using the context and a blockGasMeter for the entire block it belongs to.



Users specify how much in fees they wish to pay for the execution of their transaction. These gas meters keep track of how much gas has been used in the transaction or block so far. If the gas meter runs out, execution halts.

* **CheckTx mode:** a boolean value indicating whether a transaction should be processed in CheckTx or DeliverTx mode.
* **Min gas price:** the minimum gas price a node is willing to take to include a transaction in its block. This price is a local value configured by each node individually, and should therefore not be used in any functions in sequences leading to state transitions.
* **Consensus params:** the ABCI type Consensus Parameters, which specifies certain limits for the blockchain, such as maximum gas for a block.
* **Event manager:** allows any caller with access to a context to emit events. Modules may define module-specific events by defining various types and attributes, or by using the common definitions found in types/. Clients can subscribe or query for these events. These events are collected through DeliverTx, BeginBlock, and EndBlock and are returned to CometBFT for indexing.

[#Copy link](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#golang-context-package) Golang Context Package

A context is an immutable data structure that carries request-scoped data across APIs and processes. Contexts are also designed to enable concurrency and to be used in Go routines.



A basic context is defined in the [Golang Context Package (opens new window)↗](https://pkg.go.dev/context).

Contexts are intended to be immutable: they should never be edited. The convention is to instead create a child context from its parent using a With function. The Golang Context Package documentation instructs developers to [explicitly pass a context ctx (opens new window)↗](https://pkg.go.dev/context) as the first argument of a process.

[#Copy link](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#store-branching) Store branching

The context contains a MultiStore, which allows for branching and caching functionality using CacheMultiStore. Queries in CacheMultiStore are cached to avoid future round trips.

Each KVStore is branched in a safe and isolated ephemeral storage. Processes are free to write changes to the CacheMultiStore. If a state-transition sequence is performed without issue, the store branch can be committed to the underlying store at the end of the sequence, or it can be disregarded if something goes wrong.

[#Copy link](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#the-pattern-of-usage) The pattern of usage

The usage pattern for context is as follows:

1. Process receives a context ctx from its parent process, which provides information needed to perform the process.
2. The ctx.ms is a branched store, meaning that a branch of the multistore is made so that the process can make changes to the state as it executes without changing the original ctx.ms. This is useful to protect the underlying multistore in case the changes need to be reverted at some point in the execution.
3. The process may read and write from ctx as it is executing. It may call a subprocess and pass ctx to them as needed.
4. When a subprocess returns, it checks the result for success or failure. In case of a failure, nothing needs to be done - the branch ctx is simply discarded. If it is successful, the changes made to the CacheMultiStore can be committed to the original ctx.ms via Write().

[#Copy link](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#process) Process

Prior to calling runMsgs on any messages in the transaction, app.cacheTxContext() is used to branch and cache the context and multistore:

* For runMsgCtx, the context with the branched store is used in runMsgs to return a result.
* If the process is running in checkTxMode, there is no need to write the changes. The result is returned immediately.
* If the process is running in deliverTxMode and the result indicates a successful run over all the messages, the branched multistore is written back to the original.

****

**Show me some code for my checkers blockchain**

**Game deadline**

When a game is created or a move is played, the game needs to set its deadline some time in the future. The time it takes as *now* comes from the context.

To get this, you need a function that looks like:

Copy

func GetNextDeadline(ctx sdk.Context) time.Time {

return ctx.BlockTime().Add(MaxTurnDuration)

}

After that, it is a matter of serializing this data so it is stored alongside the other parameters of the game, and of deserializing it when checking whether it has reached the deadline.

**Gas costs**

Another point where the context is explicitly used is when you want to make your players pay with gas for operations you specify. This gas fee comes on top of the configured standard fee for transactions on your chain. Propose some ratios, which would have to be adjusted so they make sense compared to the base transaction costs:

* **Create a game:** costs **15\_000**. This should also include the costs of *closing* a game. If that was not the case, a losing player would be incentivized to let the game hit its timeout to penalize the winner.
* **Play a move:** costs **1\_000**. You could also make the final move cost zero when the player loses the game, to incentivize the player to conclude the game instead of letting it hit the timeout.
* **Reject a game:** could cost **zero**, because you want to incentivize cleaning up the state. This transaction would still cost what your chain is configured to charge for basic transactions. So you can in fact refund some gas to the player, for instance **14\_000**.

So you define the cost:

Copy

const (

CreateGameGas = 15\_000

PlayMoveGas = 1\_000

RejectGameRefundGas = 14\_000

)

Next you add the line in your MsgCreateGame handler, which already has access to the context:

Copy

func (k msgServer) CreateGame(goCtx context.Context, msg \*types.MsgCreateGame) (\*types.MsgCreateGameResponse, error) {

...

ctx.GasMeter().ConsumeGas(types.CreateGameGas, "Create game")

...

}

As for the refund when rejecting, you have to make sure that you are not trying to refund more than what was already consumed:

Copy

func (k msgServer) RejectGame(goCtx context.Context, msg \*types.MsgRejectGame) (\*types.MsgRejectGameResponse, error) {

...

refund := uint64(types.RejectGameRefundGas)

if consumed := ctx.GasMeter().GasConsumed(); consumed < refund {

refund = consumed

}

ctx.GasMeter().RefundGas(refund, "Reject game")

...

}



If you want to go beyond out-of-context code samples like the above and see in more detail how to define these features, go to [Run Your Own Cosmos Chain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/).   
  
More precisely, you can jump to:

* [Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/1-game-deadline.html), where you add the deadline feature to your chain
* [Incentivize Players](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/8-gas-meter.html), to implement gas costs

synopsis

To summarize, this section has explored:

* The importance of transaction context, which is the sum of all pertinent information about the application, the block, and the transaction itself at runtime.
* The specific properties of the context, their functions, and the processes which make use of them.
* The pattern of usage for context.
* The process which precedes running any transaction messages to branch and cache both the context and multistore.

previous

[](https://ida.interchain.io/academy/2-cosmos-concepts/10-events.html)

**[Events](https://ida.interchain.io/academy/2-cosmos-concepts/10-events.html)**

up next

**[Testing](https://ida.interchain.io/academy/2-cosmos-concepts/12-testing.html)**

[[](https://ida.interchain.io/academy/2-cosmos-concepts/12-testing.html)](https://ida.interchain.io/academy/2-cosmos-concepts/12-testing.html)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

[](https://ida.interchain.io/ida-course/discord-info.html)

On this page

[Context properties](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#context-properties)

[Golang Context Package](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#golang-context-package)

[Store branching](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#store-branching)

[The pattern of usage](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#the-pattern-of-usage)

[Process](https://ida.interchain.io/academy/2-cosmos-concepts/11-context.html#process)

#### **Get Cosmos updates**

Unsubscribe at any time. [Privacy Policy↗](https://v1.cosmos.network/privacy)

     Next

Documentation

[Cosmos SDK](https://docs.cosmos.network/)[Cosmos Hub](https://hub.cosmos.network/)[CometBFT](https://docs.cometbft.com/)[IBC Protocol](https://ibc.cosmos.network/)

Community

[Interchain blog](https://blog.cosmos.network/)[Forum](https://forum.cosmos.network/)[Discord](https://discord.gg/cosmosnetwork)

Contributing

[Source code on GitHub](https://github.com/cosmos/sdk-tutorials)

[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)

**[](https://blog.cosmos.network/)[](https://twitter.com/cosmos)[](https://discord.gg/cosmosnetwork)[](https://www.linkedin.com/company/interchain-foundation/about/)[](https://reddit.com/r/cosmosnetwork)[](https://t.me/cosmosproject)[](https://www.youtube.com/c/CosmosProject)**



Dark mode

† This website is maintained by the Interchain Foundation (ICF). The contents and opinions of this website are those of the ICF. The ICF provides links to cryptocurrency exchanges as a service to the public. The ICF does not warrant that the information provided by these websites is correct, complete, and up-to-date. The ICF is not responsible for their content and expressly rejects any liability for damages of any kind resulting from the use, reference to, or reliance on any information contained within these websites.

Cosmos is a registered trademark of the [Interchain Foundation.](https://interchain.io/)[Privacy](https://v1.cosmos.network/privacy)