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

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Getting Started](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Blockchain History](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Public and Managed Blockchains](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Consensus in Distributed Networks](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Cryptography](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Self-Assessment Quiz](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Go Introduction - First Steps](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Go Basics](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Go Interfaces](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Control Structures in Go](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Arrays and Slices in Go](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Standard Packages in Go](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Concurrency in Go](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Docker Introduction](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[The Interchain Ecosystem](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Getting ATOM and Staking It](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[A Blockchain App Architecture](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Accounts](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Transactions](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Messages](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Modules](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Protobuf](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Multistore and Keepers](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[BaseApp](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Queries](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Events](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Context](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Testing](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Relaying with IBC](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Interchain Security](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Bridges](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 2 - First Steps](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[First Steps](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Setup Your Work Environment](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Ignite CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Store Object](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Emit Game Information](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 2 Exercise](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[What is IBC?](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[IBC Token Transfer](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[IBC Tooling](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[What is CosmJS?](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Your First CosmJS Actions](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Compose Complex Transactions](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Learn to Integrate Keplr](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Auto-Expiring Games](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Handle wager payments](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Integration tests](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Incentivize Players](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Understand IBC Denoms](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Go Relayer](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Create Custom Objects](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Get an External GUI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Backend Script for Game Indexing](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[IBC Application Developer Introduction](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Create a Leaderboard Chain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Run in Production](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Prepare the Software to Run](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Prepare a Validator and Keys](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Prepare Where the Node Starts](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Prepare and Do Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Simulate Production in Docker](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Tally Player Info After Production](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Simulate a Migration in Docker](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Final Exam](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[What's Next?](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

[Continue Your Interchain Journey](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html)

Docs Version Switcher

On this page

[Some initial thoughts](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#some-initial-thoughts)

[Code needs](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#code-needs)

[New information](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#new-information)

[Additional handling](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#additional-handling)

[Unit tests](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#unit-tests)

[Integration tests](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#integration-tests)

[Adjustments](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#adjustments)

[Additional test](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#additional-test)

[Interact via the CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#interact-via-the-cli)

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#play-with-cross-chain-tokens) **Play With Cross-Chain Tokens**



Make sure you have all you need before proceeding:

* You understand the concepts of [messages](https://ida.interchain.io/academy/2-cosmos-concepts/4-messages.html), [Protobuf](https://ida.interchain.io/academy/2-cosmos-concepts/6-protobuf.html), and [IBC](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html).
* Go is installed.
* You have the checkers blockchain codebase up to the *can play* query. If not, follow some [previous steps](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html) or check out the [relevant version (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/tree/can-play-move-handler).



In this section, you will:

* Discover the Inter-Blockchain Communication Protocol.
* Accept wagers with tokens from other chains.
* Refactor unit and integration tests.

When you [introduced a wager](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/5-game-wager.html) you enabled players to play a game and bet on the outcome using the base staking token of your blockchain. What if your players want to play with *other* currencies? Your blockchain can represent a token from any other connected blockchain by using the Inter-Blockchain Communication Protocol (IBC).

Thus, you could expand the pool of your potential players by extending the pool of possible wager denominations via the use of IBC. How can you do this?



Your checkers application will be agnostic regarding tokens and relayers. Your only task is to enable the use of *foreign* tokens.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#some-initial-thoughts) Some initial thoughts

Before diving into the exercise, ask yourself:

* What new information do you need?
* How do you sanitize the inputs?
* Are there new errors to report back?
* What event should you emit?

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#code-needs) Code needs

When it comes to the code itself:

* What Ignite CLI commands, if any, assist you?
* How do you adjust what Ignite CLI created for you?
* How would you unit-test these new elements?
* How would you use Ignite CLI to locally run a one-node blockchain and interact with it via the CLI to see what you get?

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#new-information) New information

Instead of defaulting to "stake", let players decide what string represents their token:

1. Update the stored game:



Copy

message StoredGame {

...

+ string denom = 12;

}

proto /

checkers /

stored\_game.proto

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/proto/checkers/stored_game.proto" \l "L18" \t "_blank)

1. Update the message to create a game:



Copy

message MsgCreateGame {

...

+ string denom = 5;

}

proto /

checkers /

tx.proto

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/proto/checkers/tx.proto" \l "L20" \t "_blank)

1. Instruct the Ignite CLI and Protobuf to recompile both files:

**Local**

**Docker**



Copy

$ ignite generate proto-go

Copy

$ docker run --rm -it \

-v $(pwd):/checkers \

-w /checkers \

checkers\_i \

ignite generate proto-go

1. It is recommended to also update the MsgCreateGame constructor:



Copy

- func NewMsgCreateGame(creator string, black string, red string, wager uint64) \*MsgCreateGame {

+ func NewMsgCreateGame(creator string, black string, red string, wager uint64, denom string) \*MsgCreateGame {

return &MsgCreateGame{

...

+ Denom: denom,

}

}

x /

checkers /

types /

message\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/types/message_create_game.go" \l "L12-L18" \t "_blank)

1. Not to forget the CLI client:



Copy

func CmdCreateGame() \*cobra.Command {

cmd := &cobra.Command{

- Use: "create-game [black] [red] [wager]",

+ Use: "create-game [black] [red] [wager] [denom]",

Short: "Broadcast message createGame",

- Args: cobra.ExactArgs(3),

+ Args: cobra.ExactArgs(4),

RunE: func(cmd \*cobra.Command, args []string) (err error) {

...

+ argDenom := args[3]

clientCtx, err := client.GetClientTxContext(cmd)

...

msg := types.NewMsgCreateGame(

...

+ argDenom,

)

...

},

}

...

}

x /

checkers /

... /

cli /

tx\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/client/cli/tx_create_game.go" \l "L17-L39" \t "_blank)

1. This new field will be emitted during game creation, so add a new event key as a constant:



Copy

const (

...

+ GameCreatedEventDenom = "denom"

)

x /

checkers /

types /

keys.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/types/keys.go" \l "L37" \t "_blank)

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#additional-handling) Additional handling

The token denomination has been integrated into the relevant data structures. Now the proper denomination values need to be inserted in the right instances at the right locations:

1. In the helper function to create the Coin in full\_game.go:



Copy

func (storedGame \*StoredGame) GetWagerCoin() (wager sdk.Coin) {

- return sdk.NewCoin(sdk.DefaultBondDenom, sdk.NewInt(int64(storedGame.Wager)))

+ return sdk.NewCoin(storedGame.Denom, sdk.NewInt(int64(storedGame.Wager)))

}

x /

checkers /

types /

full\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/types/full_game.go" \l "L69" \t "_blank)

1. In the handler that instantiates a game:



Copy

storedGame := types.StoredGame{

...

+ Denom: msg.Denom,

}

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/keeper/msg_server_create_game.go" \l "L34" \t "_blank)

Also where it emits an event:



Copy

ctx.EventManager().EmitEvent(

sdk.NewEvent(sdk.EventTypeMessage,

...

+ sdk.NewAttribute(types.GameCreatedEventDenom, msg.Denom),

)

)

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/keeper/msg_server_create_game.go" \l "L56" \t "_blank)

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#unit-tests) Unit tests

The point of the tests is to make sure that the token denomination is correctly used. So you ought to add a denomination [when creating a game (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/keeper/msg_server_create_game_test.go#L28) and add it to [all the stored games (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/keeper/msg_server_create_game_test.go#L67) you check and all the [emitted events (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/keeper/msg_server_create_game_test.go#L121) you check. Choose a "stake" for all first games and something else for additional games, for instance ["coin" (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/keeper/msg_server_create_game_test.go#L185) and ["gold" (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/keeper/msg_server_create_game_test.go#L226) respectively.

Adjust your test helpers too:

* The coins factory now needs to care about the denomination too:



Copy

- func coinsOf(amount uint64) sdk.Coins {

+ func coinsOf(amount uint64, denom string) sdk.Coins {

return sdk.Coins{

sdk.Coin{

- Denom: sdk.DefaultBondDenom,

+ Denom: denom,

Amount: sdk.NewInt(int64(amount)),

},

}

}

x /

checkers /

testutil /

bank\_escrow\_helpers.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/testutil/bank_escrow_helpers.go" \l "L16-L19" \t "_blank)

* To minimize the amount of work to redo, add an ExpectPayWithDenom helper, and have the earlier ExpectPay use it with the "stake" denomination:



Copy

func (escrow \*MockBankEscrowKeeper) ExpectPay(context context.Context, who string, amount uint64) \*gomock.Call {

+ return escrow.ExpectPayWithDenom(context, who, amount, sdk.DefaultBondDenom)

+ }

+

+ func (escrow \*MockBankEscrowKeeper) ExpectPayWithDenom(context context.Context, who string, amount uint64, denom string) \*gomock.Call {

whoAddr, err := sdk.AccAddressFromBech32(who)

if err != nil {

panic(err)

}

- return escrow.EXPECT().SendCoinsFromAccountToModule(sdk.UnwrapSDKContext(context), whoAddr, types.ModuleName, coinsOf(amount))

+ return escrow.EXPECT().SendCoinsFromAccountToModule(sdk.UnwrapSDKContext(context), whoAddr, types.ModuleName, coinsOf(amount, denom))

}

x /

checkers /

testutil /

bank\_escrow\_helpers.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/testutil/bank_escrow_helpers.go" \l "L25-L34" \t "_blank)

Do the same with [ExpectRefund (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/testutil/bank_escrow_helpers.go#L37-L46).

With the new helpers in, you can pepper call expectations with ["coin" (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/x/checkers/keeper/end_block_server_game_test.go#L239) or "gold".

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#integration-tests) Integration tests

You have fixed your unit tests. You need to do the same for your integration tests.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#adjustments) Adjustments

You can also take this opportunity to expand the genesis state so that it includes a different coin.

* Make sure your helper to make a balance cares about the denomination:



Copy

- func makeBalance(address string, balance int64) banktypes.Balance {

+ func makeBalance(address string, balance int64, denom string) banktypes.Balance {

return banktypes.Balance{

Address: address,

Coins: sdk.Coins{

sdk.Coin{

- Denom: sdk.DefaultBondDenom,

+ Denom: denom,

Amount: sdk.NewInt(balance),

},

},

}

}

tests /

integration /

... /

keeper /

keeper\_integration\_suite\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/tests/integration/checkers/keeper/keeper_integration_suite_test.go" \l "L65-L75" \t "_blank)

* Since you want to add more coins, make a specific function to sum balances per denomination:



Copy

func addAll(balances []banktypes.Balance) sdk.Coins {

total := sdk.NewCoins()

for \_, balance := range balances {

total = total.Add(balance.Coins...)

}

return total

}

tests /

integration /

... /

keeper /

keeper\_integration\_suite\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/tests/integration/checkers/keeper/keeper_integration_suite_test.go" \l "L77-L83" \t "_blank)

* In the bank genesis creation, add new balances:



Copy

func getBankGenesis() \*banktypes.GenesisState {

coins := []banktypes.Balance{

- makeBalance(alice, balAlice),

- makeBalance(bob, balBob),

- makeBalance(carol, balCarol),

+ makeBalance(alice, balAlice, "stake"),

+ makeBalance(bob, balBob, "stake"),

+ makeBalance(bob, balBob, "coin"),

+ makeBalance(carol, balCarol, "stake"),

+ makeBalance(carol, balCarol, "coin"),

}

supply := banktypes.Supply{

- Total: coins[0].Coins.Add(coins[1].Coins...).Add(coins[2].Coins...),

+ Total: addAll(coins),

}

...

}

tests /

integration /

... /

keeper /

keeper\_integration\_suite\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/tests/integration/checkers/keeper/keeper_integration_suite_test.go" \l "L85-L95" \t "_blank)

* Also adjust the helper that checks bank balances. Add a function to reduce the amount of refactoring:



Copy

func (suite \*IntegrationTestSuite) RequireBankBalance(expected int, atAddress string) {

+ suite.RequireBankBalanceWithDenom(expected, "stake", atAddress)

+ }

+

+ func (suite \*IntegrationTestSuite) RequireBankBalanceWithDenom(expected int, denom string, atAddress string) {

sdkAdd, err := sdk.AccAddressFromBech32(atAddress)

suite.Require().Nil(err, "Failed to parse address: %s", atAddress)

suite.Require().Equal(

int64(expected),

- suite.app.BankKeeper.GetBalance(suite.ctx, sdkAdd, sdk.DefaultBondDenom).Amount.Int64())

+ suite.app.BankKeeper.GetBalance(suite.ctx, sdkAdd, denom).Amount.Int64())

}

tests /

integration /

... /

keeper /

keeper\_integration\_suite\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/tests/integration/checkers/keeper/keeper_integration_suite_test.go" \l "L110-L120" \t "_blank)

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#additional-test) Additional test

With the helpers in place, you can add a test with three players playing two games with different tokens:



Copy

func (suite \*IntegrationTestSuite) TestPlayMoveToWinnerBankPaidDifferentTokens() {

suite.setupSuiteWithOneGameForPlayMove()

goCtx := sdk.WrapSDKContext(suite.ctx)

suite.msgServer.CreateGame(goCtx, &types.MsgCreateGame{

Creator: alice,

Black: bob,

Red: carol,

Wager: 46,

Denom: "coin",

})

suite.RequireBankBalance(balAlice, alice)

suite.RequireBankBalanceWithDenom(0, "coin", alice)

suite.RequireBankBalance(balBob, bob)

suite.RequireBankBalanceWithDenom(balBob, "coin", bob)

suite.RequireBankBalance(balCarol, carol)

suite.RequireBankBalanceWithDenom(balCarol, "coin", carol)

suite.RequireBankBalance(0, checkersModuleAddress)

playAllMoves(suite.T(), suite.msgServer, sdk.WrapSDKContext(suite.ctx), "1", bob, carol, game1Moves)

playAllMoves(suite.T(), suite.msgServer, sdk.WrapSDKContext(suite.ctx), "2", bob, carol,game1Moves)

suite.RequireBankBalance(balAlice, alice)

suite.RequireBankBalanceWithDenom(0, "coin", alice)

suite.RequireBankBalance(balBob+45, bob)

suite.RequireBankBalanceWithDenom(balBob+46, "coin", bob)

suite.RequireBankBalance(balCarol-45, carol)

suite.RequireBankBalanceWithDenom(balCarol-46, "coin", carol)

suite.RequireBankBalance(0, checkersModuleAddress)

suite.RequireBankBalanceWithDenom(0, "coin", checkersModuleAddress)

}

tests /

integration /

... /

keeper /

msg\_server\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/tests/integration/checkers/keeper/msg_server_play_move_test.go" \l "L323-L350" \t "_blank)

All your tests should now pass.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#interact-via-the-cli) Interact via the CLI

Restart Ignite with chain serve. If you recall, Alice's and Bob's balances have two token denominations. Query:

**Local**

**Docker**



Copy

$ checkersd query bank balances $alice

Copy

$ docker exec -it checkers \

checkersd query bank balances $alice

This returns what you would expect from the [config.yml (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/wager-denomination/config.yml#L2-L5):



Copy

balances:

- amount: "100000000"

denom: stake

- amount: "20000"

denom: token

pagination:

next\_key: null

total: "0"

You can make use of this other token to create a new game that costs 1 token:

**Local**

**Docker**



Copy

$ checkersd tx checkers create-game \

$alice $bob 1 token \

--from $alice

Copy

$ docker exec -it checkers \

checkersd tx checkers create-game \

$alice $bob 1 token \

--from $alice

Which mentions:



Copy

...

- key: wager

value: "1"

- key: denom

value: token

...

Have Alice play once:

**Local**

**Docker**



Copy

$ checkersd tx checkers play-move 1 1 2 2 3 --from $alice

Copy

$ docker exec -it checkers \

checkersd tx checkers play-move 1 1 2 2 3 --from $alice

Which mentions:



Copy

- attributes:

- key: recipient

value: cosmos16xx0e457hm8mywdhxtmrar9t09z0mqt9x7srm3

- key: sender

value: cosmos180g0kaxzzre95f9gww93t8cqhshjydazu7g35n

- key: amount

value: 1token

type: transfer

This seems to indicate that Alice has been charged the wager. As a side node, cosmos16xx0e457hm8mywdhxtmrar9t09z0mqt9x7srm3 is the checkers module's address. Confirm it:

**Local**

**Docker**



Copy

$ checkersd query bank balances $alice

Copy

$ docker exec -it checkers \

checkersd query bank balances $alice

This returns:



Copy

balances:

- amount: "100000000"

denom: stake

- amount: "19999"

denom: token

pagination:

next\_key: null

total: "0"

Correct. You made it possible to wager any token. That includes IBC tokens.

Now check the checkers module's balance:

**Local**

**Docker**



Copy

$ checkersd query bank balances cosmos16xx0e457hm8mywdhxtmrar9t09z0mqt9x7srm3

Copy

$ docker exec -it checkers \

checkersd query bank balances cosmos16xx0e457hm8mywdhxtmrar9t09z0mqt9x7srm3

This prints:



Copy

balances:

- amount: "1"

denom: token

pagination:

next\_key: null

total: "0"

That is correct.

synopsis

To summarize, this section has explored:

* How to enable the use of cross-chain tokens to make wagers on checkers games as well as your blockchain's base staking token, by making use of the Inter-Blockchain Communication Protocol (IBC).
* How to update the stored game and the game creation message to allow players to decide what string represents their token.
* Where to insert the necessary values to allow recognition of token denominations.
* How to fix your existing tests due to the introduction of a new field and a new event, and how to add a new test when a player makes their first move.
* How to interact via the CLI to confirm the presence of the new token denomination in a player's balance and that using these tokens to make a wager functions as required.
* How to demonstrate that your application will accept IBC-foreign tokens from another blockchain, using Ignite CLI's built-in TypeScript relayer as a convenient small-scale local testing tool.

Alternatively, you can learn how to create the [TypeScript client elements](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html) for your blockchain.

previous

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

**[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)**

up next

**[Understand IBC Denoms](https://ida.interchain.io/tutorials/6-ibc-dev/)**

[[](https://ida.interchain.io/tutorials/6-ibc-dev/)](https://ida.interchain.io/tutorials/6-ibc-dev/)

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

[Some initial thoughts](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#some-initial-thoughts)

[Code needs](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#code-needs)

[New information](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#new-information)

[Additional handling](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#additional-handling)

[Unit tests](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#unit-tests)

[Integration tests](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#integration-tests)

[Adjustments](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#adjustments)

[Additional test](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#additional-test)

[Interact via the CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html#interact-via-the-cli)

#### **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)