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

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Getting Started](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Blockchain History](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Public and Managed Blockchains](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Consensus in Distributed Networks](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Cryptography](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Self-Assessment Quiz](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Go Introduction - First Steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Go Basics](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Go Interfaces](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Control Structures in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Arrays and Slices in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Standard Packages in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Concurrency in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Docker Introduction](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[The Interchain Ecosystem](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Getting ATOM and Staking It](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[A Blockchain App Architecture](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Accounts](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Transactions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Modules](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Protobuf](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Multistore and Keepers](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[BaseApp](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Queries](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Events](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Context](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Testing](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Relaying with IBC](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Interchain Security](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Bridges](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 2 - First Steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[First Steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Setup Your Work Environment](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Ignite CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Store Object](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Emit Game Information](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 2 Exercise](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[What is IBC?](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[IBC Token Transfer](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[IBC Tooling](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[What is CosmJS?](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Your First CosmJS Actions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Compose Complex Transactions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Learn to Integrate Keplr](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Auto-Expiring Games](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Handle wager payments](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Integration tests](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Incentivize Players](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Understand IBC Denoms](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Go Relayer](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Create Custom Objects](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Get an External GUI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Backend Script for Game Indexing](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[IBC Application Developer Introduction](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Create a Leaderboard Chain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Run in Production](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Prepare the Software to Run](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Prepare a Validator and Keys](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Prepare Where the Node Starts](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Prepare and Do Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Simulate Production in Docker](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Tally Player Info After Production](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Simulate a Migration in Docker](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Final Exam](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[What's Next?](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

[Continue Your Interchain Journey](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html)

Docs Version Switcher

On this page

[Some initial thoughts](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#some-initial-thoughts)

[Code needs](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#code-needs)

[Message verification coding steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#message-verification-coding-steps)

[Message handling coding steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#message-handling-coding-steps)

[Unit tests](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#unit-tests)

[Interact via the CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#interact-via-the-cli)

[Overview of upcoming content](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#overview-of-upcoming-content)

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#create-and-save-a-game-properly) **Create and Save a Game Properly**



Make sure you have everything you need before proceeding:

* You have Go installed.
* You have the checkers blockchain codebase with MsgCreateGame created by Ignite CLI. If not, follow the [previous steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/4-create-message.html) and check out [the relevant version (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/tree/create-game-msg).



In this section, you will:

* Make use of the rules of checkers.
* Update the message handler to create a game and return its ID.

In the [previous section](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/4-create-message.html), you added the message to create a game along with its serialization and dedicated gRPC function with the help of Ignite CLI.

However, it does not create a game yet because you have not implemented the message handling. How would you do this?

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#some-initial-thoughts) Some initial thoughts

Dwell on the following questions to guide you in the exercise:

* How do you sanitize your inputs?
* How do you avoid conflicts with past and future games?
* How do you use your files that implement the checkers rules?

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#code-needs) Code needs

* No Ignite CLI is involved here, it is just Go.
* Of course, you need to know where to put your code - look for TODO.
* How would you unit-test this message handling?
* 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?

For now, do not bother with niceties like gas metering or event emission.

You must add code that:

* Verifies input sanity.
* Creates a brand new game.
* Saves it in storage.
* Returns the ID of the new game.

For input sanity, your code can only accept or reject a message. You cannot *fix* a message, as that would change its content and break the signature. However, remember that your application is called via [ABCI's CheckTx](https://ida.interchain.io/academy/2-cosmos-concepts/1-architecture.html#checktx) for each transaction that it receives. It is at this point that your application can statelessly *sanitize* inputs. For each message type, Ignite CLI isolates this concern into a ValidateBasic function:



Copy

func (msg \*MsgCreateGame) ValidateBasic() error {

\_, err := sdk.AccAddressFromBech32(msg.Creator)

if err != nil {

return sdkerrors.Wrapf(sdkerrors.ErrInvalidAddress, "invalid creator address (%s)", err)

}

return nil

}

x /

checkers /

types /

message\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-msg/x/checkers/types/message_create_game.go" \l "L41-L47" \t "_blank)

It is in here that you can add further stateless checks on the message.

Ignite CLI isolated the *create a new game* concern into a separate file, x/checkers/keeper/msg\_server\_create\_game.go, for you to edit:



Copy

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

ctx := sdk.UnwrapSDKContext(goCtx)

// TODO: Handling the message

\_ = ctx

return &types.MsgCreateGameResponse{}, nil

}

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-msg/x/checkers/keeper/msg_server_create_game.go" \l "L10-L17" \t "_blank)

Ignite CLI has conveniently created all the message processing code for you. You are only required to code the key features.

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#message-verification-coding-steps) Message verification coding steps

What is a well-formatted MsgCreateGame? Eventually, you want the black and red players to be able to play moves. They will send and sign transactions for that. So, at the very least, you can check that the addresses passed are valid:



Copy

func (msg \*MsgCreateGame) ValidateBasic() error {

\_, err := sdk.AccAddressFromBech32(msg.Creator)

if err != nil {

return sdkerrors.Wrapf(sdkerrors.ErrInvalidAddress, "invalid creator address (%s)", err)

}

+ \_, err = sdk.AccAddressFromBech32(msg.Black)

+ if err != nil {

+ return sdkerrors.Wrapf(sdkerrors.ErrInvalidAddress, "invalid black address (%s)", err)

+ }

+ \_, err = sdk.AccAddressFromBech32(msg.Red)

+ if err != nil {

+ return sdkerrors.Wrapf(sdkerrors.ErrInvalidAddress, "invalid red address (%s)", err)

+ }

return nil

}

x /

checkers /

types /

message\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/types/message_create_game.go" \l "L46-L55" \t "_blank)

You should not try to check whether they have enough tokens to play as that would be a stateful check. Stateful checks are handled as part of the message handling behind ACBI's [DeliverTx](https://ida.interchain.io/academy/2-cosmos-concepts/1-architecture.html#delivertx).

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#message-handling-coding-steps) Message handling coding steps

Given that you have already done a lot of preparatory work, what coding is involved? How do you replace // TODO: Handling the message?

1. First, rules represents the ready-made file with the imported rules of the game:



Copy

import (

...

+ "github.com/alice/checkers/x/checkers/rules"

...

)

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game.go" \l "L7" \t "_blank)

1. Get the new game's ID with the [Keeper.GetSystemInfo (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/system_info.go#L17) function created by the ignite scaffold single systemInfo... command:



Copy

systemInfo, found := k.Keeper.GetSystemInfo(ctx)

if !found {

panic("SystemInfo not found")

}

newIndex := strconv.FormatUint(systemInfo.NextId, 10)

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game.go" \l "L15-L19" \t "_blank)



You panic if you cannot find the SystemInfo object because there is no way to continue if it is not there. It is not like a user error, which would warrant returning an error.

1. Create the object to be stored:



Copy

newGame := rules.New()

storedGame := types.StoredGame{

Index: newIndex,

Board: newGame.String(),

Turn: rules.PieceStrings[newGame.Turn],

Black: msg.Black,

Red: msg.Red,

}

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game.go" \l "L21-L28" \t "_blank)



Note the use of:

* + The [rules.New() (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/rules/checkers.go#L122) command, which is part of the checkers rules file you imported earlier.
  + The string content of the msg \*types.MsgCreateGame, namely .Black and .Red.

Also note that you lose the information about the creator. If your design is different, you may want to keep this information.

1. Confirm that the values in the object are correct by checking the validity of the players' addresses:



Copy

err := storedGame.Validate()

if err != nil {

return nil, err

}

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game.go" \l "L30-L33" \t "_blank)

.Red, and .Black need to be checked because they were copied as **strings**. You do not need to check .Creator because at this stage the message's signatures have been verified, and the creator is the signer.



Note that by returning an error, instead of calling panic, players cannot stall your blockchain. They can still spam but at a cost, because they will still pay the gas fee up to this point.

1. Save the StoredGame object using the [Keeper.SetStoredGame (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/stored_game.go#L10) function created by the ignite scaffold map storedGame... command:



Copy

k.Keeper.SetStoredGame(ctx, storedGame)

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game.go" \l "L35" \t "_blank)

1. Prepare the ground for the next game using the [Keeper.SetSystemInfo (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/system_info.go#L10) function created by Ignite CLI:



Copy

systemInfo.NextId++

k.Keeper.SetSystemInfo(ctx, systemInfo)

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game.go" \l "L36-L37" \t "_blank)

1. Return the newly created ID for reference:



Copy

return &types.MsgCreateGameResponse{

GameIndex: newIndex,

}, nil

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game.go" \l "L39-L41" \t "_blank)

You just handled the *create game* message by actually creating the game.

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#unit-tests) Unit tests

To test your additions to the message's ValidateBasic, you can simply add cases to the existing [message\_create\_game\_test.go (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-msg/x/checkers/types/message_create_game_test.go#L17-L28). You can verify that your additions have made the existing test fail:

**Local**

**Docker**



Copy

$ go test github.com/alice/checkers/x/checkers/types

Copy

$ docker run --rm -it \

-v $(pwd):/checkers \

-w /checkers \

checkers\_i \

go test github.com/alice/checkers/x/checkers/types

This should return:



Copy

--- FAIL: TestMsgCreateGame\_ValidateBasic (0.00s)

--- FAIL: TestMsgCreateGame\_ValidateBasic/valid\_address (0.00s)

message\_create\_game\_test.go:37:

Error Trace: /Users/alice/checkers/x/checkers/types/message\_create\_game\_test.go:37

Error: Received unexpected error:

github.com/alice/checkers/x/checkers/types.(\*MsgCreateGame).ValidateBasic

/Users/alice/checkers/x/checkers/types/message\_create\_game.go:50

github.com/alice/checkers/x/checkers/types.TestMsgCreateGame\_ValidateBasic.func1

/Users/alice/checkers/x/checkers/types/message\_create\_game\_test.go:32

invalid black address (empty address string is not allowed): invalid address

Test: TestMsgCreateGame\_ValidateBasic/valid\_address

First, change the file's package to types\_test for consistency:



Copy

- package types

+ package types\_test

import(

+ "github.com/b9lab/checkers/x/checkers/types"

)

x /

checkers /

types /

message\_create\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/types/message_create_game_test.go" \l "L1-L7" \t "_blank)

Then adjust the existing test cases and add to them:



Copy

{

- name: "invalid address",

- msg: MsgCreateGame{

+ name: "invalid creator address",

+ msg: types.MsgCreateGame{

Creator: "invalid\_address",

+ Black: sample.AccAddress(),

+ Red: sample.AccAddress(),

},

err: sdkerrors.ErrInvalidAddress,

},

+ {

+ name: "invalid black address",

+ msg: types.MsgCreateGame{

+ Creator: sample.AccAddress(),

+ Black: "invalid\_address",

+ Red: sample.AccAddress(),

+ },

+ err: sdkerrors.ErrInvalidAddress,

+ },

+ {

+ name: "invalid red address",

+ msg: types.MsgCreateGame{

+ Creator: sample.AccAddress(),

+ Black: sample.AccAddress(),

+ Red: "invalid\_address",

+ },

+ err: sdkerrors.ErrInvalidAddress,

+ },

{

- name: "valid address",

- msg: MsgCreateGame{

+ name: "valid addresses",

+ msg: types.MsgCreateGame{

Creator: sample.AccAddress(),

+ Black: sample.AccAddress(),

+ Red: sample.AccAddress(),

},

},

x /

checkers /

types /

message\_create\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/types/message_create_game_test.go" \l "L18-L52" \t "_blank)

Your tests on /types should now pass.

Moving on to the keeper, try the unit test you prepared in the previous section again:

**Local**

**Docker**



Copy

$ go test github.com/alice/checkers/x/checkers/keeper

Copy

$ docker run --rm -it \

-v $(pwd):/checkers \

-w /checkers \

checkers\_i \

go test github.com/alice/checkers/x/checkers/keeper

This should fail with:



Copy

panic: SystemInfo not found [recovered]

panic: SystemInfo not found

...

Your keeper was initialized with an empty genesis. You must fix that one way or another.

You can fix this by always initializing the keeper with the default genesis. However such a default initialization may not always be desirable. So it is better to keep this default initialization closest to the tests. Copy the setupMsgServer from [msg\_server\_test.go (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_test.go#L13-L16) into your msg\_server\_create\_game\_test.go. Modify it to also return the keeper:



Copy

func setupMsgServerCreateGame(t testing.TB) (types.MsgServer, keeper.Keeper, context.Context) {

k, ctx := keepertest.CheckersKeeper(t)

checkers.InitGenesis(ctx, \*k, \*types.DefaultGenesis())

return keeper.NewMsgServerImpl(\*k), \*k, sdk.WrapSDKContext(ctx)

}

x /

checkers /

keeper /

msg\_server\_create\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go" \l "L15-L19" \t "_blank)



Note the new import:



Copy

import (

"github.com/alice/checkers/x/checkers"

)

x /

checkers /

keeper /

msg\_server\_create\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go" \l "L8" \t "_blank)

Do not forget to replace setupMsgServer(t) with this new function everywhere in the file. For instance:



Copy

msgServer, \_, context := setupMsgServerCreateGame(t)

x /

checkers /

keeper /

msg\_server\_create\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go" \l "L22" \t "_blank)

Run the tests again with the same command as before:

**Local**

**Docker**



Copy

$ go test github.com/alice/checkers/x/checkers/keeper

Copy

$ docker run --rm -it \

-v $(pwd):/checkers \

-w /checkers \

checkers\_i \

go test github.com/alice/checkers/x/checkers/keeper

The error has changed to Not equal, and you need to adjust the expected value as per the default genesis:



Copy

require.EqualValues(t, types.MsgCreateGameResponse{

GameIndex: "1",

}, \*createResponse)

x /

checkers /

keeper /

msg\_server\_create\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go" \l "L29-L31" \t "_blank)

One unit test is good, but you can add more, in particular testing whether the values in storage are as expected when you create a single game:



Copy

func TestCreate1GameHasSaved(t \*testing.T) {

msgSrvr, keeper, context := setupMsgServerCreateGame(t)

msgSrvr.CreateGame(context, &types.MsgCreateGame{

Creator: alice,

Black: bob,

Red: carol,

})

systemInfo, found := keeper.GetSystemInfo(sdk.UnwrapSDKContext(context))

require.True(t, found)

require.EqualValues(t, types.SystemInfo{

NextId: 2,

}, systemInfo)

game1, found1 := keeper.GetStoredGame(sdk.UnwrapSDKContext(context), "1")

require.True(t, found1)

require.EqualValues(t, types.StoredGame{

Index: "1",

Board: "\*b\*b\*b\*b|b\*b\*b\*b\*|\*b\*b\*b\*b|\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*|r\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*",

Turn: "b",

Black: bob,

Red: carol,

}, game1)

}

x /

checkers /

keeper /

msg\_server\_create\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go" \l "L34-L56" \t "_blank)

Or when you [create 3 (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go#L102-L127) games. Other tests could include whether the *get all* functionality works as expected after you have created [1 game (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go#L58-L74), or [3 (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go#L181-L221), or if you create a game in a hypothetical [far future (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go#L223-L253). Also add games with [badly formatted (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go#L76-L87) or [missing input (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/create-game-handler/x/checkers/keeper/msg_server_create_game_test.go#L89-L100).

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#interact-via-the-cli) Interact via the CLI

Now you can also confirm that the transaction creates a game via the CLI. Start with:

**Local**

**Docker**



Copy

$ ignite chain serve

Copy

$ docker run --rm -it \

--name checkers \

-v $(pwd):/checkers \

-w /checkers \

checkers\_i \

ignite chain serve

Send your transaction as you did in the [previous section under "Interact via the CLI"](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/4-create-message.html):

**Local**

**Docker**



Copy

$ checkersd tx checkers create-game $alice $bob --from $alice --gas auto

Copy

$ docker exec -it checkers \

checkersd tx checkers create-game $alice $bob --from $alice --gas auto

A first good sign is that the output gas\_used is slightly higher than it was before (gas\_used: "52498"). After the transaction has been validated, confirm the current state.

Show the system info:

**Local**

**Docker**



Copy

$ checkersd query checkers show-system-info

Copy

$ docker exec -it checkers \

checkersd query checkers show-system-info

This returns:



Copy

SystemInfo:

nextId: "2"

List all stored games:

**Local**

**Docker**



Copy

$ checkersd query checkers list-stored-game

Copy

$ docker exec -it checkers \

checkersd query checkers list-stored-game

This returns a game at index 1 as expected:



Copy

pagination:

next\_key: null

total: "0"

storedGame:

- black: cosmos169mc8qqd6tlued00z23fs75tyecfcazpuwapc4

board: '\*b\*b\*b\*b|b\*b\*b\*b\*|\*b\*b\*b\*b|\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*|r\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*'

index: "1"

red: cosmos10mqyvj55hm4wunsd62wprwfv9ehcerkfghcjfl

turn: b

Show the new game alone:

**Local**

**Docker**



Copy

$ checkersd query checkers show-stored-game 1

Copy

$ docker exec -it checkers \

checkersd query checkers show-stored-game 1

This returns:



Copy

storedGame:

black: cosmos169mc8qqd6tlued00z23fs75tyecfcazpuwapc4

board: '\*b\*b\*b\*b|b\*b\*b\*b\*|\*b\*b\*b\*b|\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*|r\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*'

index: "1"

red: cosmos10mqyvj55hm4wunsd62wprwfv9ehcerkfghcjfl

turn: b

Now your game is in the blockchain's storage. Notice how alice was given the black pieces and it is already her turn to play. As a note for the next sections, this is how to understand the board:



Copy

\*b\*b\*b\*b|b\*b\*b\*b\*|\*b\*b\*b\*b|\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*|r\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*

^X:1,Y:2 ^X:3,Y:6

Or if placed in a square:



Copy

X 01234567

\*b\*b\*b\*b 0

b\*b\*b\*b\* 1

\*b\*b\*b\*b 2

\*\*\*\*\*\*\*\* 3

\*\*\*\*\*\*\*\* 4

r\*r\*r\*r\* 5

\*r\*r\*r\*r 6

r\*r\*r\*r\* 7

Y

You can also get this in a one-liner:

**Linux**

**Docker**

**Mac**



Copy

$ checkersd query checkers show-stored-game 1 --output json | jq ".storedGame.board" | sed 's/"//g' | sed 's/|/\n/g'

Copy

$ docker exec -it checkers \

bash -c "checkersd query checkers show-stored-game 1 --output json | jq \".storedGame.board\" | sed 's/\"//g' | sed 's/|/\n/g'"

Copy

$ checkersd query checkers show-stored-game 1 --output json | jq ".storedGame.board" | sed 's/"//g' | sed 's/|/\'$'\n/g'

When you are done with this exercise you can stop Ignite's chain serve.

synopsis

To summarize, this section has explored:

* How to add stateless checks on your message.
* How to implement a Message Handler that will create a new game, save it in storage, and return its ID on receiving the appropriate prompt message.
* How to create unit tests to demonstrate the validity of your code.
* How to interact via the CLI to confirm that sending the appropriate transaction will successfully create a game.

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#overview-of-upcoming-content) Overview of upcoming content

You will learn how to modify this handling in later sections by:

* Adding [new fields](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/3-game-fifo.html) to the stored information.
* Adding [an event](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html).
* Consuming [some gas](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/8-gas-meter.html).
* Facilitating the eventual [deadline enforcement](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html).
* Adding [*money*](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/5-game-wager.html) handling, including [foreign tokens](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html).

previous

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/4-create-message.html)

**[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/4-create-message.html)**

up next

**[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/6-play-game.html)**

[[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/6-play-game.html)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/6-play-game.html)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

On this page

[Some initial thoughts](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#some-initial-thoughts)

[Code needs](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#code-needs)

[Message verification coding steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#message-verification-coding-steps)

[Message handling coding steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#message-handling-coding-steps)

[Unit tests](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#unit-tests)

[Interact via the CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#interact-via-the-cli)

[Overview of upcoming content](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/5-create-handling.html#overview-of-upcoming-content)

#### **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)