[](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/9-can-play.html)



Search

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

Search



Filters

Interchain Developer Academy

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

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

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

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

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

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

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

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

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

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

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

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

[Incentivize Players](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)

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

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

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

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

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

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

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

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

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

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html)

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

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

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

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

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

Docs Version Switcher

On this page

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

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

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

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

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

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

[Interact via the CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html#interact-via-the-cli)

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



Make sure you have everything you need before proceeding:

* You understand the concepts of [queries](https://ida.interchain.io/academy/2-cosmos-concepts/9-queries.html) and [Protobuf](https://ida.interchain.io/academy/2-cosmos-concepts/6-protobuf.html).
* You have Go installed.
* You have the checkers blockchain codebase up to gas metering. If not, follow the [previous steps](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/8-gas-meter.html) or check out [the relevant version (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/tree/gas-meter).



In this section, you will:

* Improve usability with queries.
* Create a battery of unit and integration tests.

A player sends a MsgPlayMove when [making a move](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/6-play-game.html). This message can succeed or fail for several reasons. One error situation is when the message represents an invalid move. A GUI is the first place where a bad move can be caught, but it is still possible that a GUI wrongly enforces the rules.

Since sending transactions includes costs, how do you assist participants in making sure they at least do not make a wrong move?

Players would appreciate being able to confirm that a move is valid before burning gas. To add this functionality, you need to create a way for the player to call the [Move (opens new window)↗](https://github.com/batkinson/checkers-go/blob/a09daeb/checkers/checkers.go#L274) function without changing the game's state. Use a query because they are evaluated in memory and do not commit anything permanently to storage.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html#some-initial-thoughts) Some initial thoughts

When it comes to finding a correct move, ask:

* What structure will facilitate this check?
* Who do you let make such checks?
* What acceptable limitations do you have for this?
* 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/9-can-play.html#code-needs) Code needs

* What Ignite CLI commands, if any, will assist you?
* How do you adjust what Ignite CLI created for you?
* Where do you make your changes?
* 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/9-can-play.html#new-information) New information

To run a query to check the validity of a move you need to pass:

* The game ID: call the field gameIndex.
* The player color, as queries do not have a signer.
* The origin board position: fromX and fromY.
* The target board position: toX and toY.

The information to be returned is:

* A boolean for whether the move is valid, called possible.
* A text which explains why the move is not valid, called reason.

As with other data structures, you can create the query message object with Ignite CLI:

**Local**

**Docker**



Copy

$ ignite scaffold query canPlayMove \

gameIndex player fromX:uint fromY:uint toX:uint toY:uint \

--module checkers \

--response possible:bool,reason

Copy

$ docker run --rm -it \

-v $(pwd):/checkers \

-w /checkers \

checkers\_i \

ignite scaffold query canPlayMove \

gameIndex player fromX:uint fromY:uint toX:uint toY:uint \

--module checkers \

--response possible:bool,reason

Among other files, you should now have this:



Copy

message QueryCanPlayMoveRequest {

string gameIndex = 1;

string player = 2;

uint64 fromX = 3;

uint64 fromY = 4;

uint64 toX = 5;

uint64 toY = 6;

}

message QueryCanPlayMoveResponse {

bool possible = 1;

string reason = 2;

}

proto /

checkers /

query.proto

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-query/proto/checkers/query.proto" \l "L74-L86" \t "_blank)

Ignite CLI has created the following boilerplate for you:

* The [Protobuf gRPC interface function (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-query/proto/checkers/query.proto#L35-L37) to submit your new QueryCanPlayMoveRequest and its default implementation.
* The [routing of this new query (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-query/x/checkers/types/query.pb.gw.go#L424-L445) in the query facilities.
* An [empty function (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-query/x/checkers/keeper/grpc_query_can_play_move.go#L19) ready to implement the action.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html#query-handling) Query handling

Now you need to implement the answer to the player's query in grpc\_query\_can\_play\_move.go. Differentiate between two types of errors:

* Errors relating to the move, returning a reason.
* Errors indicating that testing the move is impossible, returning an error.

1. The game needs to be fetched. If it does not exist at all, you can return an error message because you did not test the move:



Copy

- // TODO: Process the query

- \_ = ctx

+ storedGame, found := k.GetStoredGame(ctx, req.GameIndex)

+ if !found {

+ return nil, sdkerrors.Wrapf(types.ErrGameNotFound, "%s", req.GameIndex)

+ }

x /

checkers /

keeper /

grpc\_query\_can\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move.go" \l "L22-L25" \t "_blank)

1. Has the game already been won?



Copy

if storedGame.Winner != rules.PieceStrings[rules.NO\_PLAYER] {

return &types.QueryCanPlayMoveResponse{

Possible: false,

Reason: types.ErrGameFinished.Error(),

}, nil

}

x /

checkers /

keeper /

grpc\_query\_can\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move.go" \l "L26-L31" \t "_blank)

1. Is the player given actually one of the game players?



Copy

isBlack := rules.PieceStrings[rules.BLACK\_PLAYER] == req.Player

isRed := rules.PieceStrings[rules.RED\_PLAYER] == req.Player

var player rules.Player

if isBlack {

player = rules.BLACK\_PLAYER

} else if isRed {

player = rules.RED\_PLAYER

} else {

return &types.QueryCanPlayMoveResponse{

Possible: false,

Reason: fmt.Sprintf("%s: %s", types.ErrCreatorNotPlayer.Error(), req.Player),

}, nil

}

x /

checkers /

keeper /

grpc\_query\_can\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move.go" \l "L32-L44" \t "_blank)

1. Is it the player's turn?



Copy

game, err := storedGame.ParseGame()

if err != nil {

return nil, err

}

if !game.TurnIs(player) {

return &types.QueryCanPlayMoveResponse{

Possible: false,

Reason: fmt.Sprintf("%s: %s", types.ErrNotPlayerTurn.Error(), player.Color),

}, nil

}

x /

checkers /

keeper /

grpc\_query\_can\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move.go" \l "L45-L54" \t "_blank)

1. Attempt the move and report back:



Copy

\_, moveErr := game.Move(

rules.Pos{

X: int(req.FromX),

Y: int(req.FromY),

},

rules.Pos{

X: int(req.ToX),

Y: int(req.ToY),

},

)

if moveErr != nil {

return &types.QueryCanPlayMoveResponse{

Possible: false,

Reason: fmt.Sprintf("%s: %s", types.ErrWrongMove.Error(), moveErr.Error()),

}, nil

}

x /

checkers /

keeper /

grpc\_query\_can\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move.go" \l "L55-L70" \t "_blank)

1. If all went well:



Copy

- return &types.QueryCanPlayMoveResponse{}, nil

+ return &types.QueryCanPlayMoveResponse{

+ Possible: true,

+ Reason: "ok",

+ }, nil

x /

checkers /

keeper /

grpc\_query\_can\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move.go" \l "L72-L75" \t "_blank)

Quite straightforward.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html#unit-tests) Unit tests

A query is evaluated in memory, while using the current state in a read-only mode. Thanks to this, you can take some liberties with the current state before running a test, as long as reading the state works as intended. For example, you can pretend that the game has been progressed through a number of moves even though you have only just planted the board in that state in the keeper. For this reason, you can easily test the new method with unit tests, even though you painstakingly prepared integration tests.

Take inspiration from [the other tests on queries (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_system_info_test.go#L20-L35), which create an array of cases to test in a loop. Running a battery of test cases makes it easier to insert new cases and surface any unintended impact. Create a new grpc\_query\_can\_play\_move\_test.go file where you:

1. Declare a struct that describes a test case:



Copy

type canPlayGameCase struct {

desc string

game types.StoredGame

request \*types.QueryCanPlayMoveRequest

response \*types.QueryCanPlayMoveResponse

err string

}

x /

checkers /

keeper /

grpc\_query\_can\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move_test.go" \l "L12-L18" \t "_blank)

1. Create the common OK response, so as to reuse it:



Copy

var (

canPlayOkResponse = &types.QueryCanPlayMoveResponse{

Possible: true,

Reason: "ok",

}

)

x /

checkers /

keeper /

grpc\_query\_can\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move_test.go" \l "L21-L24" \t "_blank)

1. Prepare your array of cases:



Copy

canPlayTestRange = []canPlayGameCase{

// TODO

}

x /

checkers /

keeper /

grpc\_query\_can\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move_test.go" \l "L25" \t "_blank)

1. In the array add your first test case, one that returns an OK response:



Copy

{

desc: "First move by black",

game: 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",

Winner: "\*",

},

request: &types.QueryCanPlayMoveRequest{

GameIndex: "1",

Player: "b",

FromX: 1,

FromY: 2,

ToX: 2,

ToY: 3,

},

response: canPlayOkResponse,

err: "nil",

},

x /

checkers /

keeper /

grpc\_query\_can\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move_test.go" \l "L26-L44" \t "_blank)

1. Add [other test cases (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move_test.go#L26-L239). Examples include a missing request:



Copy

{

desc: "Nil request, wrong",

game: 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",

Winner: "\*",

},

request: nil,

response: nil,

err: "rpc error: code = InvalidArgument desc = invalid request",

},

x /

checkers /

keeper /

grpc\_query\_can\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move_test.go" \l "L45-L56" \t "_blank)

Or a player playing out of turn:



Copy

{

desc: "First move by red, wrong",

game: 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",

Winner: "\*",

},

request: &types.QueryCanPlayMoveRequest{

GameIndex: "1",

Player: "r",

FromX: 1,

FromY: 2,

ToX: 2,

ToY: 3,

},

response: &types.QueryCanPlayMoveResponse{

Possible: false,

Reason: "player tried to play out of turn: red",

},

err: "nil",

},

x /

checkers /

keeper /

grpc\_query\_can\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move_test.go" \l "L139-L160" \t "_blank)

1. With the test cases defined, add a single test function that runs all the cases:



Copy

func TestCanPlayCasesAsExpected(t \*testing.T) {

for \_, testCase := range canPlayTestRange {

keeper, ctx := keepertest.CheckersKeeper(t)

goCtx := sdk.WrapSDKContext(ctx)

t.Run(testCase.desc, func(t \*testing.T) {

keeper.SetStoredGame(ctx, testCase.game)

response, err := keeper.CanPlayMove(goCtx, testCase.request)

if testCase.response == nil {

require.Nil(t, response)

} else {

require.EqualValues(t, testCase.response, response)

}

if testCase.err == "nil" {

require.Nil(t, err)

} else {

require.EqualError(t, err, testCase.err)

}

})

}

}

x /

checkers /

keeper /

grpc\_query\_can\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/x/checkers/keeper/grpc_query_can_play_move_test.go" \l "L243-L262" \t "_blank)



All test cases are run within a single unit test. To avoid having one case bleed into the next, the keeper is created afresh inside the loop.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html#integration-tests) Integration tests

You can also add integration tests on top of your unit tests. Put them alongside your other integration tests. Create grpc\_query\_can\_play\_move\_test.go.

Test if it is possible to play on the first game that is created in the system:



Copy

func (suite \*IntegrationTestSuite) TestCanPlayAfterCreate() {

suite.setupSuiteWithOneGameForPlayMove()

goCtx := sdk.WrapSDKContext(suite.ctx)

response, err := suite.queryClient.CanPlayMove(goCtx, &types.QueryCanPlayMoveRequest{

GameIndex: "1",

Player: "b",

FromX: 1,

FromY: 2,

ToX: 2,

ToY: 3,

})

suite.Require().Nil(err)

suite.Require().EqualValues(canPlayOkResponse, response)

}

tests /

integration /

... /

keeper /

grpc\_query\_can\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/can-play-move-handler/tests/integration/checkers/keeper/grpc_query_can_play_move_test.go" \l "L13-L26" \t "_blank)

With these, your query handling function should be covered.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/9-can-play.html#interact-via-the-cli) Interact via the CLI

Set the game expiry to 5 minutes and start ignite chain serve. Remember that the CLI can always inform you about available commands:

**Local**

**Docker**



Copy

$ checkersd query checkers --help

Copy

$ docker exec -it checkers \

checkersd query checkers --help

Which prints:



Copy

...

Available Commands:

can-play-move Query canPlayMove

...

What can checkersd tell you about the command:

**Local**

**Docker**



Copy

$ checkersd query checkers can-play-move --help

Copy

$ docker exec -it checkers \

checkersd query checkers can-play-move --help

Which prints:



Copy

...

Usage:

checkersd query checkers can-play-move [gameIndex] [player] [fromX] [fromY] [toX] [toY] [flags]

...

You can test this query at any point in a game's life.

1

When there is no such game:

**Local**

**Docker**



Copy

$ checkersd query checkers can-play-move 2048 r 1 2 2 3

Copy

$ docker exec -it checkers \

checkersd query checkers can-play-move 2048 r 1 2 2 3

Trying this on a game that does not exist returns:



Copy

Error: rpc error: code = InvalidArgument desc = 2048: game by id not found: invalid request

...

Confirm this was an error from the point of view of the executable:



Copy

$ echo $?

This prints:



Copy

1

There is room to improve the error message, but it is important that you got an error, as expected.

2

When you ask for a bad player color:

**Local**

**Docker**



Copy

$ checkersd tx checkers create-game \

$alice $bob 1000000 \

--from $alice -y

$ checkersd query checkers can-play-move 1 w 1 2 2 3

Copy

$ docker exec -it checkers \

checkersd tx checkers create-game \

$alice $bob 1000000 \

--from $alice -y

$ docker exec -it checkers \

checkersd query checkers can-play-move 1 w 1 2 2 3

If the player tries to play the wrong color on a game that exists, it returns:



Copy

possible: false

reason: 'message creator is not a player: w'

This is a proper message response, and a reason elaborating on the message.

3

When you ask for a player out of turn:

**Local**

**Docker**



Copy

$ checkersd query checkers can-play-move 1 r 0 5 1 4

Copy

$ docker exec -it checkers \

checkersd query checkers can-play-move 1 r 0 5 1 4

If the opponent tries to play out of turn, it returns:



Copy

possible: false

reason: 'player tried to play out of turn: red'

4

When you ask for a piece that is not that of the player:

**Local**

**Docker**



Copy

$ checkersd query checkers can-play-move 1 b 0 5 1 4

Copy

$ docker exec -it checkers \

checkersd query checkers can-play-move 1 b 0 5 1 4

If black tries to play a red piece, it returns:



Copy

possible: false

reason: 'wrong move: Not {red}''s turn'

5

When it is correct:

**Local**

**Docker**



Copy

$ checkersd query checkers can-play-move 1 b 1 2 2 3

Copy

$ docker exec -it checkers \

checkersd query checkers can-play-move 1 b 1 2 2 3

If black tests a correct move, it returns:



Copy

possible: true

reason: ok

6

When the player must capture:

**Local**

**Docker**



Copy

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

$ checkersd tx checkers play-move 1 0 5 1 4 --from $bob -y

$ checkersd query checkers can-play-move 1 b 2 3 3 4

Copy

$ docker exec -it checkers \

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

$ docker exec -it checkers \

checkersd tx checkers play-move 1 0 5 1 4 --from $bob -y

$ docker exec -it checkers \

checkersd query checkers can-play-move 1 b 2 3 3 4

If black fails to capture a mandatory red piece, it returns:



Copy

possible: false

reason: 'wrong move: Invalid move: {2 3} to {3 4}'

The reason given is understandable, but it does not clarify why the move is invalid. There is room to improve this message.

7

After the game has been forfeited:

**Local**

**Docker**



Copy

$ checkersd tx checkers create-game \

$alice $bob 1000000 \

--from $alice -y

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

$ checkersd tx checkers play-move 2 0 5 1 4 --from $bob -y

$ checkersd query checkers can-play-move 2 b 2 3 0 5

Copy

$ docker exec -it checkers \

checkersd tx checkers create-game \

$alice $bob 1000000 \

--from $alice -y

$ docker exec -it checkers \

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

$ docker exec -it checkers \

checkersd tx checkers play-move 2 0 5 1 4 --from $bob -y

$ docker exec -it checkers \

checkersd query checkers can-play-move 2 b 2 3 0 5

If black tries to capture a red piece on a running game, it returns:



Copy

possible: true

reason: ok

Wait five minutes for the forfeit:

**Local**

**Docker**



Copy

$ checkersd query checkers can-play-move 2 b 2 3 0 5

Copy

$ docker exec -it checkers \

checkersd query checkers can-play-move 2 b 2 3 0 5

Now it returns:



Copy

possible: false

reason: game is already finished

These query results satisfy our expectations.

synopsis

To summarize, this section has explored:

* How application usability can be improved with queries, such as by avoiding the cost of sending *technically* valid transactions which will nevertheless inevitably be rejected due to the application's current state.
* How queries allow the user to evaluate the application state in read-only mode, without committing anything permanently to storage, with the result that a planned transaction can be judged as acceptable or not before burning gas.
* How effective query construction will allow the application to signal not just that a planned transaction will fail but also the *reason* it will fail, improving the user's knowledge base for future actions.
* How to create a query object with Ignite CLI; implement appropriate answers to a player's query; perform integration tests which extrapolate on the application's actual current state; and interact via the CLI to test the effectiveness of the query object.

previous

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/8-gas-meter.html)

**[Incentivize Players](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/8-gas-meter.html)**

up next

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

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

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

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

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

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

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

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