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

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Getting Started](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Blockchain History](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Public and Managed Blockchains](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Consensus in Distributed Networks](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Cryptography](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Self-Assessment Quiz](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Go Introduction - First Steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Go Basics](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Go Interfaces](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Control Structures in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Arrays and Slices in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Standard Packages in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Concurrency in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Docker Introduction](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[The Interchain Ecosystem](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Getting ATOM and Staking It](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[A Blockchain App Architecture](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Accounts](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Transactions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Modules](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Protobuf](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Multistore and Keepers](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[BaseApp](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Queries](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Events](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Context](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Testing](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Relaying with IBC](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Interchain Security](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Bridges](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Week 2 - First Steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[First Steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Setup Your Work Environment](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Ignite CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Store Object](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Emit Game Information](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Week 2 Exercise](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[What is IBC?](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[IBC Token Transfer](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[IBC Tooling](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[What is CosmJS?](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Your First CosmJS Actions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Compose Complex Transactions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Learn to Integrate Keplr](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Auto-Expiring Games](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Handle wager payments](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Integration tests](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Incentivize Players](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Understand IBC Denoms](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Go Relayer](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Create Custom Objects](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Get an External GUI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Backend Script for Game Indexing](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[IBC Application Developer Introduction](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Create a Leaderboard Chain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

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

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

[Run in Production](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Prepare the Software to Run](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Prepare a Validator and Keys](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Prepare Where the Node Starts](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Prepare and Do Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Simulate Production in Docker](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Tally Player Info After Production](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Simulate a Migration in Docker](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Final Exam](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[What's Next?](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

[Continue Your Interchain Journey](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

Docs Version Switcher

On this page

[Some initial thoughts](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#some-initial-thoughts)

[Code needs](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#code-needs)

[Game-created event](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#game-created-event)

[Player-moved event](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#player-moved-event)

[Unit tests](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#unit-tests)

[Interact with the CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#interact-with-the-cli)

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#emit-game-information) **Emit Game Information**



Make sure you have everything you need before proceeding:

* You understand the concepts of [events](https://ida.interchain.io/academy/2-cosmos-concepts/10-events.html).
* Go is installed.
* You have the checkers blockchain codebase with MsgPlayMove and its handling. If not, follow the [previous steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/6-play-game.html) or check out [the relevant version (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/tree/play-move-handler).



In this section, you will:

* Define event types.
* Emit events.
* Extend unit tests.

Now that you have [added the possible actions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/6-play-game.html), including their return values, use events to notify players. Your blockchain can now create and play games. However, it does not inform the outside world about this in a convenient way. That is where events come in - but what do you need to emit them?

Imagine a potential or current player waiting for their turn. It is not practical to look at all the transactions and search for the ones signifying the player's turn. It is better to listen to known events that let clients determine which player's turn it is.

Adding events to your application is as simple as:

1. Defining the events you want to use.
2. Emitting corresponding events as actions unfold.

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#some-initial-thoughts) Some initial thoughts

Before you dive into the specifics of the exercise, ask yourself:

* Why do actions warrant a detailed event?
* What level of detail goes into each event?
* How do you make it easy for external parties to understand your events?
* At what stage do you emit events?

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#code-needs) Code needs

Now start by thinking about the following:

* How do you adjust your code to do all this?
* 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?

Only focus on the narrow issue of event emission.

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#game-created-event) Game-created event

Start with the event that announces the creation of a new game. The goal is to:

* Inform the players about the game.
* Make it easy for the players to find the relevant game.

Define new keys in x/checkers/types/keys.go:



Copy

const (

GameCreatedEventType = "new-game-created" // Indicates what event type to listen to

GameCreatedEventCreator = "creator" // Subsidiary information

GameCreatedEventGameIndex = "game-index" // What game is relevant

GameCreatedEventBlack = "black" // Is it relevant to me?

GameCreatedEventRed = "red" // Is it relevant to me?

)

x /

checkers /

types /

keys.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/two-events/x/checkers/types/keys.go" \l "L28-L34" \t "_blank)

Emit the event in your handler file x/checkers/keeper/msg\_server\_create\_game.go:



Copy

ctx.EventManager().EmitEvent(

sdk.NewEvent(types.GameCreatedEventType,

sdk.NewAttribute(types.GameCreatedEventCreator, msg.Creator),

sdk.NewAttribute(types.GameCreatedEventGameIndex, newIndex),

sdk.NewAttribute(types.GameCreatedEventBlack, msg.Black),

sdk.NewAttribute(types.GameCreatedEventRed, msg.Red),

),

)

x /

checkers /

keeper /

msg\_server\_create\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/two-events/x/checkers/keeper/msg_server_create_game.go" \l "L39-L46" \t "_blank)

Now you must implement this correspondingly in the GUI, or include a server to listen for such events.

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#player-moved-event) Player-moved event

The created transaction to play a move informs the opponent about:

* Which player is relevant.
* Which game the move relates to.
* When the move happened.
* The move's outcome.
* Whether the game was won.



Contrary to the *create game* event, which alerted the players about a new game, the players now know which game IDs to watch for. There is no need to repeat the players' addresses, the game ID is information enough.

You define new keys in x/checkers/types/keys.go similarly:



Copy

const (

MovePlayedEventType = "move-played"

MovePlayedEventCreator = "creator"

MovePlayedEventGameIndex = "game-index"

MovePlayedEventCapturedX = "captured-x"

MovePlayedEventCapturedY = "captured-y"

MovePlayedEventWinner = "winner"

)

x /

checkers /

types /

keys.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/two-events/x/checkers/types/keys.go" \l "L36-L43" \t "_blank)

Emit the event in your file x/checkers/keeper/msg\_server\_play\_move.go:



Copy

ctx.EventManager().EmitEvent(

sdk.NewEvent(types.MovePlayedEventType,

sdk.NewAttribute(types.MovePlayedEventCreator, msg.Creator),

sdk.NewAttribute(types.MovePlayedEventGameIndex, msg.GameIndex),

sdk.NewAttribute(types.MovePlayedEventCapturedX, strconv.FormatInt(int64(captured.X), 10)),

sdk.NewAttribute(types.MovePlayedEventCapturedY, strconv.FormatInt(int64(captured.Y), 10)),

sdk.NewAttribute(types.MovePlayedEventWinner, rules.PieceStrings[game.Winner()]),

),

)

x /

checkers /

keeper /

msg\_server\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/two-events/x/checkers/keeper/msg_server_play_move.go" \l "L61-L69" \t "_blank)

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#unit-tests) Unit tests

The unit tests you have created so far still pass. However you also want to confirm that the events have been emitted in both situations. The events are recorded in the context, so the test is a little bit different. In msg\_server\_create\_game\_test.go, add this test:



Copy

func TestCreate1GameEmitted(t \*testing.T) {

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

msgSrvr.CreateGame(context, &types.MsgCreateGame{

Creator: alice,

Black: bob,

Red: carol,

})

ctx := sdk.UnwrapSDKContext(context)

require.NotNil(t, ctx)

events := sdk.StringifyEvents(ctx.EventManager().ABCIEvents())

require.Len(t, events, 1)

event := events[0]

require.EqualValues(t, sdk.StringEvent{

Type: "new-game-created",

Attributes: []sdk.Attribute{

{Key: "creator", Value: alice},

{Key: "game-index", Value: "1"},

{Key: "black", Value: bob},

{Key: "red", Value: carol},

},

}, event)

}

x /

checkers /

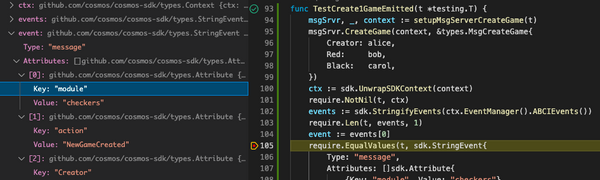
keeper /

msg\_server\_create\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/two-events/x/checkers/keeper/msg_server_create_game_test.go" \l "L76-L97" \t "_blank)

How can you *guess* the order of elements? Easily, as you created them in this order. Alternatively, you can *peek* by using Visual Studio Code:

1. Put a breakpoint on the line after event := events[0].
2. Run this test in **debug mode**: right-click the green arrow next to the test name.
3. Observe the live values on the left.



As for the events emitted during the *play move* test, there are two of them: one for the creation and the other for the play. Because this is a unit test and each action is not isolated into individual transactions, the context collects all events emitted during the test. It just so happens that the context prepends them - the newest one is at index 0. Which is why, when you fetch them, the play event is at events[0].



Copy

func TestPlayMoveEmitted(t \*testing.T) {

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

msgServer.PlayMove(context, &types.MsgPlayMove{

Creator: bob,

GameIndex: "1",

FromX: 1,

FromY: 2,

ToX: 2,

ToY: 3,

})

ctx := sdk.UnwrapSDKContext(context)

require.NotNil(t, ctx)

events := sdk.StringifyEvents(ctx.EventManager().ABCIEvents())

require.Len(t, events, 2)

event := events[0]

require.EqualValues(t, sdk.StringEvent{

Type: "move-played",

Attributes: []sdk.Attribute{

{Key: "creator", Value: bob},

{Key: "game-index", Value: "1"},

{Key: "captured-x", Value: "-1"},

{Key: "captured-y", Value: "-1"},

{Key: "winner", Value: "\*"},

},

}, event)

}

x /

checkers /

keeper /

msg\_server\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/two-events/x/checkers/keeper/msg_server_play_move_test.go" \l "L110-L135" \t "_blank)

When two players play one after the other, the context collates the attributes of move-played all together in a single array in an appending fashion, with the older attributes at the lower indices, starting at 0. For instance, you have to rely on array slices like event.Attributes[5:] to test the attributes of the second move-played event:



Copy

func TestPlayMove2Emitted(t \*testing.T) {

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

msgServer.PlayMove(context, &types.MsgPlayMove{

Creator: bob,

GameIndex: "1",

FromX: 1,

FromY: 2,

ToX: 2,

ToY: 3,

})

msgServer.PlayMove(context, &types.MsgPlayMove{

Creator: carol,

GameIndex: "1",

FromX: 0,

FromY: 5,

ToX: 1,

ToY: 4,

})

ctx := sdk.UnwrapSDKContext(context)

require.NotNil(t, ctx)

events := sdk.StringifyEvents(ctx.EventManager().ABCIEvents())

require.Len(t, events, 2)

event := events[0]

require.Equal(t, "move-played", event.Type)

require.EqualValues(t, []sdk.Attribute{

{Key: "creator", Value: carol},

{Key: "game-index", Value: "1"},

{Key: "captured-x", Value: "-1"},

{Key: "captured-y", Value: "-1"},

{Key: "winner", Value: "\*"},

}, event.Attributes[5:])

}

x /

checkers /

keeper /

msg\_server\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/two-events/x/checkers/keeper/msg_server_play_move_test.go" \l "L261-L292" \t "_blank)

Try these tests:

**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

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#interact-with-the-cli) Interact with the CLI

If you did not do it already, start your chain with Ignite.

Alice made a move. Will Bob's move emit an event?

**Local**

**Docker**



Copy

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

Copy

$ docker exec -it checkers \

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

The log is longer and not very readable, but the expected elements are present:



Copy

...

raw\_log: '[{"events":[{"type":"message","attributes":[{"key":"action","value":"play\_move"}]},{"type":"move-played","attributes":[{"key":"creator","value":"cosmos1xf6s64kaw7at7um8lnwj65vadxqr6hnyhr9v83"},{"key":"game-index","value":"1"},{"key":"captured-x","value":"-1"},{"key":"captured-y","value":"-1"},{"key":"winner","value":"\*"}]}]}]'

To parse the events and display them in a more user-friendly way, take the txhash again:



Copy

$ checkersd query tx 531E5708A1EFBE08D14ABF947FBC888BFC69CD6F04A589D478204BF3BA891AB7 --output json | jq ".raw\_log | fromjson"

**Local**

**Docker**



Copy

$ checkersd query tx 531E5708A1EFBE08D14ABF947FBC888BFC69CD6F04A589D478204BF3BA891AB7 --output json | jq '.raw\_log | fromjson'

Copy

$ docker exec -it checkers \

bash -c "checkersd query tx 531E5708A1EFBE08D14ABF947FBC888BFC69CD6F04A589D478204BF3BA891AB7 --output json | jq '.raw\_log | fromjson'"

This returns something like:



Copy

[

{

"events": [

{

"type": "message",

"attributes": [

{

"key": "action",

"value": "play\_move"

}

]

},

{

"type": "move-played",

"attributes": [

{

"key": "creator",

"value": "cosmos1xf6s64kaw7at7um8lnwj65vadxqr6hnyhr9v83"

},

{

"key": "game-index",

"value": "1"

},

{

"key": "captured-x",

"value": "-1"

},

{

"key": "captured-y",

"value": "-1"

},

{

"key": "winner",

"value": "\*"

}

]

}

]

}

]

As you can see, no pieces were captured. However, it turns out that Bob placed his piece ready to be captured by Alice:

**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'

Which prints:



Copy

\*b\*b\*b\*b

b\*b\*b\*b\*

\*\*\*b\*b\*b

\*\*b\*\*\*\*\*

\*r\*\*\*\*\*\* <-- Ready to be captured

\*\*r\*r\*r\*

\*r\*r\*r\*r

r\*r\*r\*r\*

The rules of the game included in this project mandate that the player captures a piece when possible. So Alice captures the piece:

**Local**

**Docker**



Copy

$ checkersd tx checkers play-move 1 2 3 0 5 --from $alice

Copy

$ docker exec -it checkers \

checkersd tx checkers play-move 1 2 3 0 5 --from $alice

This returns:



Copy

...

raw\_log: '[{"events":[{"type":"message","attributes":[{"key":"action","value":"play\_move"}]},{"type":"move-played","attributes":[{"key":"creator","value":"cosmos1qxeu0aclpl45429aeveh3t4e7y9ghr22r5d9r2"},{"key":"game-index","value":"1"},{"key":"captured-x","value":"1"},{"key":"captured-y","value":"4"},{"key":"winner","value":"\*"}]}]}]'

When formatted for clarity, you see the following::



Copy

[

{

"events": [

{

"type": "message",

"attributes": [

{

"key": "action",

"value": "play\_move"

}

]

},

{

"type": "move-played",

"attributes": [

{

"key": "creator",

"value": "cosmos1qxeu0aclpl45429aeveh3t4e7y9ghr22r5d9r2"

},

{

"key": "game-index",

"value": "1"

},

{

"key": "captured-x",

"value": "1"

},

{

"key": "captured-y",

"value": "4"

},

{

"key": "winner",

"value": "\*"

}

]

}

]

}

]

Correct: Alice captured a piece and the board now looks like this:



Copy

\*b\*b\*b\*b

b\*b\*b\*b\*

\*\*\*b\*b\*b

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

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

\*r\*r\*r\*r

r\*r\*r\*r\*

This confirms that the *play* event is emitted as expected. You can confirm the same for the *game created* event.

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

synopsis

To summarize, this section has explored:

* How to define event types and then emit events to cause the UI to notify players of game actions as they occur, such as creating games and playing moves.
* How listening to known events which let clients determine which player must move next is better than the impractical alternative of examining all transactions to search for the ones which signify a player's turn.
* How to define a Game-created event that will notify the participating players and make it easy for them to find the game.
* How to define a Player-moved event that will indicate which player and game is involved, when the move occurred, the move's outcome, and whether the game was won as a result.
* How to test your code to ensure that it functions as desired.
* How to interact with the CLI to check the effectiveness of an emitted event.

previous

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

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

up next

**[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)**

[[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.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/1-ignite-cli/7-events.html#some-initial-thoughts)

[Code needs](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#code-needs)

[Game-created event](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#game-created-event)

[Player-moved event](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#player-moved-event)

[Unit tests](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#unit-tests)

[Interact with the CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html#interact-with-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)