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

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Getting Started](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Blockchain History](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Public and Managed Blockchains](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Consensus in Distributed Networks](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Cryptography](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Self-Assessment Quiz](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Go Introduction - First Steps](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Go Basics](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Go Interfaces](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Control Structures in Go](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Arrays and Slices in Go](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Standard Packages in Go](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Concurrency in Go](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Docker Introduction](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[The Interchain Ecosystem](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Getting ATOM and Staking It](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[A Blockchain App Architecture](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Accounts](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Transactions](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Messages](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Modules](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Protobuf](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Multistore and Keepers](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[BaseApp](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Queries](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Events](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Context](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Testing](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Relaying with IBC](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Interchain Security](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Bridges](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Migrations](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 2 - First Steps](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[First Steps](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Setup Your Work Environment](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Ignite CLI](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Store Object](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Emit Game Information](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 2 Exercise](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[What is IBC?](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[IBC Token Transfer](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[IBC Tooling](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[What is CosmJS?](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Your First CosmJS Actions](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Compose Complex Transactions](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Learn to Integrate Keplr](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Auto-Expiring Games](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Handle wager payments](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Integration tests](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Incentivize Players](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Understand IBC Denoms](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Go Relayer](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Create Custom Objects](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Get an External GUI](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Backend Script for Game Indexing](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[IBC Application Developer Introduction](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Create a Leaderboard Chain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Run in Production](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Prepare the Software to Run](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Prepare a Validator and Keys](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Prepare Where the Node Starts](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Prepare and Do Migrations](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Simulate Production in Docker](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Tally Player Info After Production](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Simulate a Migration in Docker](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Final Exam](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[What's Next?](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

[Continue Your Interchain Journey](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html)

Docs Version Switcher

On this page

[Scaffold a packet with Ignite CLI](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#scaffold-a-packet-with-ignite-cli)

[Proto definitions](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#proto-definitions)

[CLI commands](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#cli-commands)

[SendPacket and packet callback logic](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#sendpacket-and-packet-callback-logic)

[Sending packets](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#sending-packets)

[Receiving packets](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#receiving-packets)

[Acknowledging packets](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#acknowledging-packets)

[Timing out packets](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#timing-out-packets)

[Extra details](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#extra-details)

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#adding-packet-and-acknowledgement-data) **Adding Packet and Acknowledgement Data**

This section demonstrates how to define packets and acks (acknowledgements) for the leaderboard blockchain.



This blockchain will mostly be receiving packets from the checkers blockchain or other gaming chains, not sending them. This will be handled in the [checkers blockchain extension section](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/7-ibc-app-leaderboard.html).

In this section, you will add an additional packet definition that will enable the Leaderboard chain to send a packet to connected game chains when a player has entered the top of the rankings. It is up to you if you want to include this in the Leaderboard chain you will build in the checkers extension exercise.

The documentation on how to define packets and acks in the Inter-Blockchain Communication Protocol (IBC) can be found in [the ibc-go docs (opens new window)↗](https://ibc.cosmos.network/main/ibc/apps/packets_acks.html).

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#scaffold-a-packet-with-ignite-cli) Scaffold a packet with Ignite CLI

You are now going to scaffold the IBC packet data with Ignite CLI, and compare it once more with *git diff*:

**Local**

**Docker**



Copy

$ ignite scaffold packet ibcTopRank \

playerId rank:uint score:uint \

--ack playerId \

--module leaderboard

Copy

$ docker run --rm -it \

-v $(pwd):/leaderboard \

-w /leaderboard \

ignitehq/cli:0.22.1 \

scaffold packet ibcTopRank \

playerId rank:uint score:uint \

--ack playerId \

--module leaderboard



The packet is called ibcTopRank, which includes the fields playerId, rank, and score. Additionally, you send back the playerId of the player who entered the top of the rankings through the Acknowledgement.

The output on the terminal gives an overview of the changes made:



Copy

modify proto/leaderboard/packet.proto

modify proto/leaderboard/tx.proto

modify x/leaderboard/client/cli/tx.go

create x/leaderboard/client/cli/tx\_ibc\_top\_rank.go

modify x/leaderboard/handler.go

create x/leaderboard/keeper/ibc\_top\_rank.go

create x/leaderboard/keeper/msg\_server\_ibc\_top\_rank.go

modify x/leaderboard/module\_ibc.go

modify x/leaderboard/types/codec.go

modify x/leaderboard/types/events\_ibc.go

create x/leaderboard/types/messages\_ibc\_top\_rank.go

create x/leaderboard/types/messages\_ibc\_top\_rank\_test.go

create x/leaderboard/types/packet\_ibc\_top\_rank.go

🎉 Created a packet `ibcTopRank`.

In the next paragraphs, you will investigate each of the most important additions to the code.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#proto-definitions) Proto definitions

The first additions are to the proto definitions in the packet.proto and tx.proto files:



Copy

message LeaderboardPacketData {

oneof packet {

NoData noData = 1;

// this line is used by starport scaffolding # ibc/packet/proto/field

+ IbcTopRankPacketData ibcTopRankPacket = 2;

// this line is used by starport scaffolding # ibc/packet/proto/field/number

}

}

separate /

leaderboard /

... /

leaderboard /

packet.proto

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/proto/leaderboard/packet.proto" \l "L10" \t "_blank)

One addition is IbcTopRankPacketData:



Copy

// IbcTopRankPacketData defines a struct for the packet payload

message IbcTopRankPacketData {

string playerId = 1;

uint64 rank = 2;

uint64 score = 3;

}

separate /

leaderboard /

... /

leaderboard /

packet.proto

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/proto/leaderboard/packet.proto" \l "L19-L24" \t "_blank)

The next addition is the ack:



Copy

// IbcTopRankPacketAck defines a struct for the packet acknowledgement

message IbcTopRankPacketAck {

string playerId = 1;

}

separate /

leaderboard /

... /

leaderboard /

packet.proto

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/proto/leaderboard/packet.proto" \l "L27-L29" \t "_blank)

And in tx.proto a Message service is added:



Copy

// Msg defines the Msg service.

service Msg {

+ rpc SendIbcTopRank(MsgSendIbcTopRank) returns (MsgSendIbcTopRankResponse);

// this line is used by starport scaffolding # proto/tx/rpc

}

separate /

leaderboard /

... /

leaderboard /

tx.proto

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/proto/leaderboard/tx.proto" \l "L11" \t "_blank)

Where:



Copy

message MsgSendIbcTopRank {

string creator = 1;

string port = 2;

string channelID = 3;

uint64 timeoutTimestamp = 4;

string playerId = 5;

uint64 rank = 6;

uint64 score = 7;

}

message MsgSendIbcTopRankResponse {

}

separate /

leaderboard /

... /

leaderboard /

tx.proto

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/proto/leaderboard/tx.proto" \l "L15-L27" \t "_blank)



The proto message MsgSendIbcTopRank includes the field timeoutTimestamp, which is added by Ignite CLI when scaffolding an IBC packet. This is an IBC channel parameter that is important in IBC and Ignite CLI abstracts this away, removing the need for the user to add this manually.



The proto definitions will be compiled into types/packet.pb.go and types/tx.pb.go.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#cli-commands) CLI commands

Ignite CLI also creates CLI commands to send packets and adds them to the client/cli/ folder.

Packets can be sent from the CLI with the following command:



Copy

$ leaderboardd tx leaderboard send-ibc-top-rank \

[portID] [channelID] [playerId] [rank] [score]

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#sendpacket-and-packet-callback-logic) SendPacket and packet callback logic

When scaffolding an IBC module with Ignite CLI, you already saw the implementation of the IBCModule interface, including a bare-bones packet callbacks structure. Now that you have also scaffolded a packet (and ack), the callbacks have been added with logic to handle the receive, ack, and timeout scenarios.

Additionally, for the sending of a packet, a message server has been added that handles a SendPacket message, in this case MsgSendIbcTopRank.



IBC allows some freedom to the developers regarding how to implement the custom logic, decoding and encoding packets, and processing acks. The provided structure is but one example of how to tackle this. Therefore, it makes sense to focus on the general flow to handle user messages or IBC callbacks rather than the specific implementation by Ignite CLI.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#sending-packets) Sending packets

To handle a user submitting a message to send an IBC packet, a message server is added to the handler:



Copy

func NewHandler(k keeper.Keeper) sdk.Handler {

+ msgServer := keeper.NewMsgServerImpl(k)

return func(ctx sdk.Context, msg sdk.Msg) (\*sdk.Result, error) {

ctx = ctx.WithEventManager(sdk.NewEventManager())

switch msg := msg.(type) {

+ case \*types.MsgSendIbcTopRank:

+ res, err := msgServer.SendIbcTopRank(sdk.WrapSDKContext(ctx), msg)

+ return sdk.WrapServiceResult(ctx, res, err)

// this line is used by starport scaffolding # 1

default:

errMsg := fmt.Sprintf("unrecognized %s message type: %T", types.ModuleName, msg)

return nil, sdkerrors.Wrap(sdkerrors.ErrUnknownRequest, errMsg)

}

}

}

separate /

leaderboard /

... /

leaderboard /

handler.go

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/handler.go" \l "L13-L30" \t "_blank)

It calls the SendIbcTopRank method on the message server, defined as:



Copy

func (k msgServer) SendIbcTopRank(goCtx context.Context, msg \*types.MsgSendIbcTopRank) (\*types.MsgSendIbcTopRankResponse, error) {

ctx := sdk.UnwrapSDKContext(goCtx)

// TODO: logic before transmitting the packet

// Construct the packet

var packet types.IbcTopRankPacketData

packet.PlayerId = msg.PlayerId

packet.Rank = msg.Rank

packet.Score = msg.Score

// Transmit the packet

err := k.TransmitIbcTopRankPacket(

ctx,

packet,

msg.Port,

msg.ChannelID,

clienttypes.ZeroHeight(),

msg.TimeoutTimestamp,

)

if err != nil {

return nil, err

}

return &types.MsgSendIbcTopRankResponse{}, nil

}

separate /

leaderboard /

... /

keeper /

msg\_server\_ibc\_top\_rank.go

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/keeper/msg_server_ibc_top_rank.go" \l "L11" \t "_blank)

This in turn calls the TransmitIbcTopRankPacket method on the module's keeper, defined in x/leaderboard/keeper/ibc\_top\_rank.go. This method gets all of the required metadata from core IBC before sending the packet using the ChannelKeeper's SendPacket function:



Copy

func (k Keeper) TransmitIbcTopRankPacket(

ctx sdk.Context,

packetData types.IbcTopRankPacketData,

sourcePort,

sourceChannel string,

timeoutHeight clienttypes.Height,

timeoutTimestamp uint64,

) error {

sourceChannelEnd, found := k.ChannelKeeper.GetChannel(ctx, sourcePort, sourceChannel)

... // error validation

destinationPort := sourceChannelEnd.GetCounterparty().GetPortID()

destinationChannel := sourceChannelEnd.GetCounterparty().GetChannelID()

// get the next sequence

sequence, found := k.ChannelKeeper.GetNextSequenceSend(ctx, sourcePort, sourceChannel)

... // error validation

channelCap, ok := k.ScopedKeeper.GetCapability(ctx, host.ChannelCapabilityPath(sourcePort, sourceChannel))

... // error validation

packetBytes, err := packetData.GetBytes()

... // error validation

packet := channeltypes.NewPacket(

packetBytes,

sequence,

sourcePort,

sourceChannel,

destinationPort,

destinationChannel,

timeoutHeight,

timeoutTimestamp,

)

if err := k.ChannelKeeper.SendPacket(ctx, channelCap, packet); err != nil {

return err

}

return nil

}

separate /

leaderboard /

... /

keeper /

ibc\_top\_rank.go

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/keeper/ibc_top_rank.go" \l "L15" \t "_blank)



When you want to add additional custom logic before transmitting the packet, you do this in the SendIbcTopRank method on the message server.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#receiving-packets) Receiving packets

In a previous section you examined the OnRecvPacket callback in the x/leaderboard/module\_ibc.go file. There, Ignite CLI had set up a structure to dispatch the packet depending on packet type through a switch statement. Now by adding the IbcTopRank packet, a case has been added:



Copy

// @ switch packet := modulePacketData.Packet.(type) in OnRecvPacket

case \*types.LeaderboardPacketData\_IbcTopRankPacket:

packetAck, err := am.keeper.OnRecvIbcTopRankPacket(ctx, modulePacket, \*packet.IbcTopRankPacket)

if err != nil {

ack = channeltypes.NewErrorAcknowledgement(err.Error())

} else {

// Encode packet acknowledgement

packetAckBytes, err := types.ModuleCdc.MarshalJSON(&packetAck)

if err != nil {

return channeltypes.NewErrorAcknowledgement(sdkerrors.Wrap(sdkerrors.ErrJSONMarshal, err.Error()).Error())

}

ack = channeltypes.NewResultAcknowledgement(sdk.MustSortJSON(packetAckBytes))

}

ctx.EventManager().EmitEvent(

sdk.NewEvent(

types.EventTypeIbcTopRankPacket,

sdk.NewAttribute(sdk.AttributeKeyModule, types.ModuleName),

sdk.NewAttribute(types.AttributeKeyAckSuccess, fmt.Sprintf("%t", err != nil)),

),

)

separate /

leaderboard /

... /

leaderboard /

module\_ibc.go

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/module_ibc.go" \l "L144" \t "_blank)

The first line of code in the case statement calls the application's OnRecvIbcTopRankPacket callback on the keeper to process the reception of the packet:



Copy

// OnRecvIbcTopRankPacket processes packet reception

func (k Keeper) OnRecvIbcTopRankPacket(ctx sdk.Context, packet channeltypes.Packet, data types.IbcTopRankPacketData) (packetAck types.IbcTopRankPacketAck, err error) {

// validate packet data upon receiving

if err := data.ValidateBasic(); err != nil {

return packetAck, err

}

// TODO: packet reception logic

return packetAck, nil

}

separate /

leaderboard /

... /

keeper /

ibc\_top\_rank.go

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/keeper/ibc_top_rank.go" \l "L70" \t "_blank)



Remember that the OnRecvPacket callback writes an acknowledgement as well (this course covers the synchronous write ack case).

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#acknowledging-packets) Acknowledging packets

Similarly to the OnRecvPacket case before, Ignite CLI has already prepared the structure of the OnAcknowledgementPacket with the switch statement. Again, scaffolding the packet adds a case to the switch:



Copy

// @ switch packet := modulePacketData.Packet.(type) in OnAcknowledgementPacket

case \*types.LeaderboardPacketData\_IbcTopRankPacket:

err := am.keeper.OnAcknowledgementIbcTopRankPacket(ctx, modulePacket, \*packet.IbcTopRankPacket, ack)

if err != nil {

return err

}

eventType = types.EventTypeIbcTopRankPacket

This calls into the newly created application keeper's ack packet callback:



Copy

func (k Keeper) OnAcknowledgementIbcTopRankPacket(ctx sdk.Context, packet channeltypes.Packet, data types.IbcTopRankPacketData, ack channeltypes.Acknowledgement) error {

switch dispatchedAck := ack.Response.(type) {

case \*channeltypes.Acknowledgement\_Error:

// TODO: failed acknowledgement logic

\_ = dispatchedAck.Error

return nil

case \*channeltypes.Acknowledgement\_Result:

// Decode the packet acknowledgement

var packetAck types.IbcTopRankPacketAck

if err := types.ModuleCdc.UnmarshalJSON(dispatchedAck.Result, &packetAck); err != nil {

// The counter-party module doesn't implement the correct acknowledgement format

return errors.New("cannot unmarshal acknowledgement")

}

// TODO: successful acknowledgement logic

return nil

default:

// The counter-party module doesn't implement the correct acknowledgement format

return errors.New("invalid acknowledgement format")

}

}

separate /

leaderboard /

... /

keeper /

ibc\_top\_rank.go

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/keeper/ibc_top_rank.go" \l "L119" \t "_blank)

This allows us to add custom application logic for both failed and successful acks.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#timing-out-packets) Timing out packets

Timing out the packets follows the same flow, adding a case to the switch statement in OnTimeoutPacket, calling into the keeper's timeout packet callback where the custom logic can be implemented. It is left to the reader to investigate this independently.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#extra-details) Extra details

Next to the above, some additions have also been made to the types package. These include [codec.go (opens new window)↗](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/types/codec.go), [events\_ibc.go (opens new window)↗](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/types/events_ibc.go), and [messages\_ibc\_top\_rank.go (opens new window)↗](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/types/messages_ibc_top_rank.go).

Again, the reader is invited to check these out independently.



Events in IBC are important because relayers process events to check if there are packets (or acknowledgements) to relay.   
  
Ignite CLI has scaffolded some events in x/leaderboard/types/events\_ibc.go for timeout and the ibcTopRank packet which you have defined:



Copy

package types

// IBC events

const (

EventTypeTimeout = "timeout"

EventTypeIbcTopRankPacket = "ibcTopRank\_packet"

// this line is used by starport scaffolding # ibc/packet/event

AttributeKeyAckSuccess = "success"

AttributeKeyAck = "acknowledgement"

AttributeKeyAckError = "error"

)

separate /

leaderboard /

... /

types /

events\_ibc.go

[View source→](https://github.com/b9lab/cosmos-ibc-docker/blob/main/separate/leaderboard/ida-content/x/leaderboard/types/events_ibc.go" \t "_blank)

Here are found both the Event type and the attributes it contains.   
  
These are not the only relevant events for IBC, though, the others can be found in the core IBC source code:

* [Client events (opens new window)↗](https://github.com/cosmos/ibc-go/blob/main/modules/core/02-client/types/events.go)
* [Connection events (opens new window)↗](https://github.com/cosmos/ibc-go/blob/main/modules/core/03-connection/types/events.go)
* [Channel events (opens new window)↗](https://github.com/cosmos/ibc-go/blob/main/modules/core/04-channel/types/events.go)

You can go back to the code examined so far to take note of the events emitted.

synopsis

To summarize, this section has explored:

* Scaffolding a chain with an IBC-enabled module, with both chain and module called leaderboard.
* How Ignite CLI made sure to implement the IBCModule interface, including channel handshake and packet callbacks.
* How Ignite CLI has bound our IBC module to a port and added a route to the IBC router.
* Scaffolding an IBC packet, IbcTopRankPacket.
* How Ignite CLI defined the packet and ack data.
* How Ignite CLI sets up the basic message handling and packet handling to send, receive, acknowledge, and timeout packets.



Even though the ability to send and receive packets is now enabled, no application logic to execute has yet been implemented. This is outside the scope of this section. The reader is invited to follow the [checkers blockchain extension exercise](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/6-ibc-app-checkers.html).



When scaffolding a packet, Ignite CLI will ensure the chain can act both as the sender or receiver of a packet by default. This is a symmetrical setup which makes sense for some applications, like ICS20.   
  
However, it is also possible to have an asymmetrical setup where one chain will always be the source *or* destination chain for a given packet, not both. In this case, the message server and packet callbacks can be updated to error when, for example, a chain receives a packet though it is supposed to exclusively be the destination chain. Interchain Accounts or ICS27 is an example of this asymmetrical situation, as is the checkers extension exercise.

previous

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/4-ibc-app-steps.html)

**[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/4-ibc-app-steps.html)**

up next

**[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/6-ibc-app-checkers.html)**

[[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/6-ibc-app-checkers.html)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/6-ibc-app-checkers.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

[Scaffold a packet with Ignite CLI](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#scaffold-a-packet-with-ignite-cli)

[Proto definitions](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#proto-definitions)

[CLI commands](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#cli-commands)

[SendPacket and packet callback logic](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#sendpacket-and-packet-callback-logic)

[Sending packets](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#sending-packets)

[Receiving packets](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#receiving-packets)

[Acknowledging packets](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#acknowledging-packets)

[Timing out packets](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#timing-out-packets)

[Extra details](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/5-ibc-app-packets.html#extra-details)

#### **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)