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

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Getting Started](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Blockchain History](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Public and Managed Blockchains](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Consensus in Distributed Networks](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Cryptography](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Self-Assessment Quiz](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Go Introduction - First Steps](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Go Basics](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Go Interfaces](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Control Structures in Go](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Arrays and Slices in Go](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Standard Packages in Go](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Concurrency in Go](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Docker Introduction](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[The Interchain Ecosystem](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Getting ATOM and Staking It](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[A Blockchain App Architecture](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Accounts](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Transactions](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Messages](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Modules](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Protobuf](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Multistore and Keepers](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[BaseApp](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Queries](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Events](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Context](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Testing](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Relaying with IBC](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Interchain Security](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Bridges](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Migrations](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 2 - First Steps](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[First Steps](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Setup Your Work Environment](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Ignite CLI](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Store Object](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Emit Game Information](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 2 Exercise](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[What is IBC?](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[IBC Token Transfer](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[IBC Tooling](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[What is CosmJS?](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Your First CosmJS Actions](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Compose Complex Transactions](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Learn to Integrate Keplr](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Auto-Expiring Games](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Handle wager payments](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Integration tests](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Incentivize Players](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Understand IBC Denoms](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Go Relayer](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Create Custom Objects](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Get an External GUI](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Backend Script for Game Indexing](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[IBC Application Developer Introduction](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Create a Leaderboard Chain](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Run in Production](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Prepare the Software to Run](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Prepare a Validator and Keys](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Prepare Where the Node Starts](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Prepare and Do Migrations](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Simulate Production in Docker](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Tally Player Info After Production](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Simulate a Migration in Docker](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Final Exam](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[What's Next?](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

[Continue Your Interchain Journey](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html)

Docs Version Switcher

On this page

[Compile Protobuf](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#compile-protobuf)

[Prepare integration](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#prepare-integration)

[Integration tests](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#integration-tests)

[Preparation](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#preparation)

[First tests](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#first-tests)

[Prepare your checkers chain](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#prepare-your-checkers-chain)

[Launch the tests](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#launch-the-tests)

[A note on Docker networks](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#a-note-on-docker-networks)

[#Copy link](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#create-custom-objects) **Create Custom Objects**



Make sure you have everything you need before proceeding:

* You understand the concepts of [Protobuf](https://ida.interchain.io/academy/2-cosmos-concepts/6-protobuf.html).
* You have completed the introductory [CosmJS tutorial](https://ida.interchain.io/tutorials/7-cosmjs/1-cosmjs-intro.html).
* Go and npm are installed.
* You have finished the checkers Go blockchain exercise. If not, you can follow the tutorial [here](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/10-wager-denom.html), or just clone and checkout the [relevant branch (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/tree/wager-denomination) that contains the final version.

With your checkers application ready for use, it is a good time to prepare client elements that eventually allow you to create a GUI and/or server-side scripts. Here, you will apply [what you have learned](https://ida.interchain.io/tutorials/7-cosmjs/5-create-custom.html) about creating your own custom CosmJS interfaces.

Before you can get into working on your application directly, you need to make sure CosmJS understands your checkers module and knows how to interact with it. This generally means you need to create the Protobuf objects and clients in TypeScript and create extensions that facilitate the use of them.

[#Copy link](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#compile-protobuf) Compile Protobuf

You will have to create a client folder that will contain all these new elements. If you want to keep the Go parts of your checkers project separate from the TypeScript parts, you can use another repository for the *client*. To keep a link between the two repositories, add the *client* parts as a submodule to your Go parts:



Copy

$ git submodule add git@github.com:cosmos/academy-checkers-ui.git client

Replace the path with your own repository. In effect, this creates a new client folder. This client folder makes it possible for you to easily update another repository with content generated out of your Go code.

Create a folder named scripts in your project root. This is where you will launch the Protobuf compilation:



Copy

$ mkdir -p scripts/protoc

In the scripts folder, or in your Docker image, install a compiler:

**Local**

**Docker**



Copy

$ cd scripts/protoc

$ curl -L https://github.com/protocolbuffers/protobuf/releases/download/v21.5/protoc-21.5-linux-x86\_64.zip -o protoc.zip

$ unzip protoc.zip

$ rm protoc.zip

# If /usr/local/bin is in your $PATH

$ ln -s $(pwd)/bin/protoc /usr/local/bin/protoc

$ cd ../..

Make sure that:

* /usr/local/bin is the right folder to link to in the command above.
* You are downloading the right executable for your computer; see your options [here (opens new window)↗](https://github.com/protocolbuffers/protobuf/releases/tag/v21.5).

Copy

- FROM --platform=linux ubuntu:22.04

+ FROM --platform=linux ubuntu:22.04 as base

ARG BUILDARCH

...

ENV MOCKGEN\_VERSION=1.6.0

+ ENV PROTOC\_VERSION=21.7

+

+ FROM base AS platform-amd64

+ ENV PROTOC\_PLATFORM=x86\_64

+

+ FROM base AS platform-arm64

+ ENV PROTOC\_PLATFORM=aarch\_64

+

+ FROM platform-${BUILDARCH}

ENV LOCAL=/usr/local

...

- ENV PACKAGES curl gcc jq make

+ ENV PACKAGES curl gcc jq make unzip

...

RUN go install github.com/golang/mock/mockgen@v${MOCKGEN\_VERSION}

+ # Install ProtoC

+ RUN mkdir -p /usr/lib/protoc

+ WORKDIR /usr/lib/protoc

+ RUN curl -L https://github.com/protocolbuffers/protobuf/releases/download/v${PROTOC\_VERSION}/protoc-${PROTOC\_VERSION}-linux-${PROTOC\_PLATFORM}.zip -o protoc.zip

+ RUN unzip -o protoc.zip

+ RUN rm protoc.zip

+ RUN ln -s /usr/lib/protoc/bin/protoc ${LOCAL}/bin/protoc

...

Dockerfile-ubuntu

[View source](https://github.com/cosmos/b9-checkers-academy-draft/blob/cosmjs-elements/Dockerfile-ubuntu" \l "L1-L50" \t "_blank)

Rebuild your Docker image.

Now install your additional modules:

**Local**

**Docker**



Copy

$ cd scripts

$ npm install ts-proto@1.121.6 --save-dev --save-exact

$ cd ..

Copy

$ docker run --rm -it \

-v $(pwd):/checkers \

-w /checkers/scripts \

checkers\_i \

npm install ts-proto@1.121.6 --save-dev --save-exact

Create the folder structure to receive the compiled files:



Copy

$ mkdir -p client/src/types/generated

Check what Cosmos SDK version you are using:



Copy

$ grep cosmos-sdk go.mod

This may return:



Copy

github.com/cosmos/cosmos-sdk v0.45.4

Download the required files from your .proto files:



Copy

$ mkdir -p proto/cosmos/base/query/v1beta1

$ curl https://raw.githubusercontent.com/cosmos/cosmos-sdk/v0.45.4/proto/cosmos/base/query/v1beta1/pagination.proto -o proto/cosmos/base/query/v1beta1/pagination.proto

$ mkdir -p proto/google/api

$ curl https://raw.githubusercontent.com/cosmos/cosmos-sdk/v0.45.4/third\_party/proto/google/api/annotations.proto -o proto/google/api/annotations.proto

$ curl https://raw.githubusercontent.com/cosmos/cosmos-sdk/v0.45.4/third\_party/proto/google/api/http.proto -o proto/google/api/http.proto

$ mkdir -p proto/gogoproto

$ curl https://raw.githubusercontent.com/cosmos/cosmos-sdk/v0.45.4/third\_party/proto/gogoproto/gogo.proto -o proto/gogoproto/gogo.proto

Now compile:

**Local**

**Docker**



Copy

$ cd scripts

$ ls ../proto/checkers | xargs -I {} protoc \

--plugin="./node\_modules/.bin/protoc-gen-ts\_proto" \

--ts\_proto\_out="../client/src/types/generated" \

--proto\_path="../proto" \

--ts\_proto\_opt="esModuleInterop=true,forceLong=long,useOptionals=messages" \

checkers/{}

Copy

$ ls proto/checkers | xargs -I {} docker run --rm \

-v $(pwd):/checkers \

-w /checkers/scripts \

checkers\_i \

protoc \

--plugin="./node\_modules/.bin/protoc-gen-ts\_proto" \

--ts\_proto\_out="../client/src/types/generated" \

--proto\_path="../proto" \

--ts\_proto\_opt="esModuleInterop=true,forceLong=long,useOptionals=messages" \

checkers/{}

You should now have your TypeScript files.

In order to easily repeat these steps in the future, you can add them to your existing Makefile with slight modifications:



Copy

install-protoc-gen-ts:

mkdir -p scripts/protoc

cd scripts && npm install

curl -L https://github.com/protocolbuffers/protobuf/releases/download/v21.5/protoc-21.5-linux-x86\_64.zip -o scripts/protoc/protoc.zip

cd scripts/protoc && unzip -o protoc.zip

rm scripts/protoc/protoc.zip

ln -s $(pwd)/scripts/protoc/bin/protoc /usr/local/bin/protoc

cosmos-version = v0.45.4

download-cosmos-proto:

mkdir -p proto/cosmos/base/query/v1beta1

curl https://raw.githubusercontent.com/cosmos/cosmos-sdk/${cosmos-version}/proto/cosmos/base/query/v1beta1/pagination.proto -o proto/cosmos/base/query/v1beta1/pagination.proto

mkdir -p proto/google/api

curl https://raw.githubusercontent.com/cosmos/cosmos-sdk/${cosmos-version}/third\_party/proto/google/api/annotations.proto -o proto/google/api/annotations.proto

curl https://raw.githubusercontent.com/cosmos/cosmos-sdk/${cosmos-version}/third\_party/proto/google/api/http.proto -o proto/google/api/http.proto

mkdir -p proto/gogoproto

curl https://raw.githubusercontent.com/cosmos/cosmos-sdk/${cosmos-version}/third\_party/proto/gogoproto/gogo.proto -o proto/gogoproto/gogo.proto

gen-protoc-ts: download-cosmos-proto install-protoc-gen-ts

mkdir -p ./client/src/types/generated/

ls proto/checkers | xargs -I {} protoc \

--plugin="./scripts/node\_modules/.bin/protoc-gen-ts\_proto" \

--ts\_proto\_out="./client/src/types/generated" \

--proto\_path="./proto" \

--ts\_proto\_opt="esModuleInterop=true,forceLong=long,useOptionals=messages" \

checkers/{}

Makefile

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/cosmjs-elements/Makefile" \l "L6-L31" \t "_blank)

Whenever you want to re-compile them, run:

**Local**

**Docker**



Copy

$ make gen-protoc-ts

Copy

$ docker run --rm \

-v $(pwd):/checkers \

-w /checkers \

checkers\_i \

make gen-protoc-ts

You have created the [basic Protobuf objects (opens new window)↗](https://github.com/cosmos/academy-checkers-ui/tree/generated/src/types/generated/checkers) that will assist you with communicating with the blockchain.

[#Copy link](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#prepare-integration) Prepare integration

At this point, you have the generated files in your client folder. If you have made this client folder as a Git submodule, then you can work directly in it and do not need to go back to the checkers Cosmos SDK:



Copy

$ cd client

Also, if you use Docker and did not go through the trouble of building the Docker image for the checkers Cosmos SDK, you can use the node:18.7-slim image.

Install the Protobuf.js package in your client project:

**Local**

**Docker**



Copy

$ npm install protobufjs@7.0.0 --save-exact

Copy

$ docker run --rm \

-v $(pwd):/client \

-w /client \

node:18.7-slim \

npm install protobufjs@7.0.0 --save-exact

At a later stage, you will add checkers as an extension to Stargate, but you can define your checkers extension immediately. The canPlay query could make use of better types for player and position. Start by declaring them in client/src/checkers/player.ts:



Copy

export type Player = "b" | "r"

export type GamePiece = Player | "\*"

export interface Pos {

x: number

y: number

}

src /

types /

checkers /

player.ts

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/src/types/checkers/player.ts" \l "L1-L6" \t "_blank)

Your checkers extension will need to use the CosmJS Stargate package. Install it:

**Local**

**Docker**



Copy

$ npm install @cosmjs/stargate@0.28.11 --save-exact

Copy

$ docker run --rm \

-v $(pwd):/client \

-w /client \

node:18.7-slim \

npm install @cosmjs/stargate@0.28.11 --save-exact

Now you can declare the checkers extension in src/modules/checkers/queries.ts:



Copy

export interface AllStoredGameResponse {

storedGames: StoredGame[]

pagination?: PageResponse

}

export interface CheckersExtension {

readonly checkers: {

readonly getSystemInfo: () => Promise<SystemInfo>

readonly getStoredGame: (index: string) => Promise<StoredGame | undefined>

readonly getAllStoredGames: (

key: Uint8Array,

offset: Long,

limit: Long,

countTotal: boolean,

) => Promise<AllStoredGameResponse>

readonly canPlayMove: (

index: string,

player: Player,

from: Pos,

to: Pos,

) => Promise<QueryCanPlayMoveResponse>

}

}

src /

modules /

checkers /

queries.ts

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/src/modules/checkers/queries.ts" \l "L15-L37" \t "_blank)

Do not forget a *setup* function, as this is expected by Stargate:



Copy

export function setupCheckersExtension(base: QueryClient): CheckersExtension {

const rpc = createProtobufRpcClient(base)

// Use this service to get easy typed access to query methods

// This cannot be used for proof verification

const queryService = new QueryClientImpl(rpc)

return {

checkers: {

getSystemInfo: async (): Promise<SystemInfo> => {

const { SystemInfo } = await queryService.SystemInfo({})

assert(SystemInfo)

return SystemInfo

},

getStoredGame: async (index: string): Promise<StoredGame | undefined> => {

const response: QueryGetStoredGameResponse = await queryService.StoredGame({

index: index,

})

return response.storedGame

},

getAllStoredGames: async (

key: Uint8Array,

offset: Long,

limit: Long,

countTotal: boolean,

): Promise<AllStoredGameResponse> => {

const response: QueryAllStoredGameResponse = await queryService.StoredGameAll({

pagination: {

key: key,

offset: offset,

limit: limit,

countTotal: countTotal,

reverse: false,

},

})

return {

storedGames: response.storedGame,

pagination: response.pagination,

}

},

canPlayMove: async (

index: string,

player: Player,

from: Pos,

to: Pos,

): Promise<QueryCanPlayMoveResponse> => {

return queryService.CanPlayMove({

gameIndex: index,

player: player,

fromX: Long.fromNumber(from.x),

fromY: Long.fromNumber(from.y),

toX: Long.fromNumber(to.x),

toY: Long.fromNumber(to.y),

})

},

},

}

}

Expand



src /

modules /

checkers /

queries.ts

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/src/modules/checkers/queries.ts" \l "L39-L95" \t "_blank)

You may have to add these imports by hand:



Copy

import { assert } from "@cosmjs/utils"

import Long from "long"

src /

modules /

checkers /

queries.ts

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/src/modules/checkers/queries.ts" \l "L2-L3" \t "_blank)

Now create your CheckersStargateClient in src/checkers\_stargateclient.ts:



Copy

export class CheckersStargateClient extends StargateClient {

public readonly checkersQueryClient: CheckersExtension | undefined

public static async connect(

endpoint: string,

options?: StargateClientOptions,

): Promise<CheckersStargateClient> {

const tmClient = await Tendermint34Client.connect(endpoint)

return new CheckersStargateClient(tmClient, options)

}

protected constructor(tmClient: Tendermint34Client | undefined, options: StargateClientOptions = {}) {

super(tmClient, options)

if (tmClient) {

this.checkersQueryClient = QueryClient.withExtensions(tmClient, setupCheckersExtension)

}

}

}

src /

checkers\_stargateclient.ts

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/src/checkers_stargateclient.ts" \l "L5-L22" \t "_blank)

[#Copy link](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#integration-tests) Integration tests

It is possible to already run some integration tests against a running checkers blockchain.

[#Copy link](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#preparation) Preparation

Install packages to run tests.

**Local**

**Docker**



Copy

$ npm install mocha@10.0.0 @types/mocha@9.1.1 \

chai@4.3.6 @types/chai@4.3.3 \

ts-node@10.9.1 @types/node@18.7.5 \

dotenv@16.0.1 @types/dotenv@8.2.0 \

--save-dev --save-exact

Copy

$ docker run --rm \

-v $(pwd):/client \

-w /client \

node:18.7-slim \

npm install mocha@10.0.0 @types/mocha@9.1.1 \

chai@4.3.6 @types/chai@4.3.3 \

ts-node@10.9.1 @types/node@18.7.5 \

dotenv@16.0.1 @types/dotenv@8.2.0 \

--save-dev --save-exact

Describe how to connect to the running blockchain in a .env file in your project root. This depends on where you will run the tests, not on where you run the blockchain:

**Local**

**Docker**



Copy

RPC\_URL="http://localhost:26657"

.env

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/.env" \l "L1" \t "_blank)

Copy

RPC\_URL="http://checkers:26657"

.env

[View source](https://github.com/cosmos/academy-checkers-ui/blob/stargate/.env" \l "L1" \t "_blank)

This will run the checkers chain in a container named checkers.

Alternatively, use whichever address connects to the RPC port of the checkers blockchain.

This information will be picked up by the dotenv package. Now let TypeScript know about this in an environment.d.ts file:



Copy

declare global {

namespace NodeJS {

interface ProcessEnv {

RPC\_URL: string

}

}

}

export {}

environment.d.ts

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/environment.d.ts" \t "_blank)

Also add your tconfig.json as you see fit:



Copy

{

"exclude": ["./tests/", "./node\_modules/", "./dist/"],

"compilerOptions": {

"esModuleInterop": true,

"module": "ES2015",

"moduleResolution": "node",

"target": "ES6"

}

}

tsconfig.json

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/tsconfig.json" \t "_blank)

Add the line that describes how the tests are run:



Copy

{

...

"scripts": {

- "test": "echo \"Error: no test specified\" && exit 1"

+ "test": "env TS\_NODE\_COMPILER\_OPTIONS='{\"module\": \"commonjs\" }' mocha --require ts-node/register 'test/\*\*/\*.ts'"

},

...

}

package.json

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/package.json" \l "L7" \t "_blank)

[#Copy link](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#first-tests) First tests

Because the intention is to run these tests against a running chain, possibly a fresh one, they cannot expect too much, such as how many games have been created so far. Still, it is possible to at least test that the connection is made and queries pass through.

Create test/integration/system-info.ts:



Copy

import { expect } from "chai"

import { config } from "dotenv"

import \_ from "../../environment"

import { CheckersStargateClient } from "../../src/checkers\_stargateclient"

import { CheckersExtension } from "../../src/modules/checkers/queries"

config()

describe("SystemInfo", function () {

let client: CheckersStargateClient, checkers: CheckersExtension["checkers"]

before("create client", async function () {

client = await CheckersStargateClient.connect(process.env.RPC\_URL)

checkers = client.checkersQueryClient!.checkers

})

it("can get system info", async function () {

const systemInfo = await checkers.getSystemInfo()

expect(systemInfo.nextId.toNumber()).to.be.greaterThanOrEqual(1)

expect(parseInt(systemInfo.fifoHeadIndex, 10)).to.be.greaterThanOrEqual(-1)

expect(parseInt(systemInfo.fifoTailIndex, 10)).to.be.greaterThanOrEqual(-1)

})

})

test /

integration /

system-info.ts

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/test/integration/system-info.ts" \t "_blank)

And create one for stored games:



Copy

import { expect } from "chai"

import { config } from "dotenv"

import Long from "long"

import \_ from "../../environment"

import { CheckersStargateClient } from "../../src/checkers\_stargateclient"

import { CheckersExtension } from "../../src/modules/checkers/queries"

config()

describe("StoredGame", function () {

let client: CheckersStargateClient, checkers: CheckersExtension["checkers"]

before("create client", async function () {

client = await CheckersStargateClient.connect(process.env.RPC\_URL)

checkers = client.checkersQueryClient!.checkers

})

it("can get game list", async function () {

const allGames = await checkers.getAllStoredGames(

Uint8Array.of(),

Long.fromInt(0),

Long.fromInt(0),

true,

)

expect(allGames.storedGames).to.be.length.greaterThanOrEqual(0)

})

it("cannot get non-existent game", async function () {

try {

await checkers.getStoredGame("no-id")

expect.fail("It should have failed")

} catch (error) {

expect(error.toString()).to.equal(

"Error: Query failed with (22): rpc error: code = NotFound desc = not found: key not found",

)

}

})

})

test /

integration /

stored-game.ts

[View source→](https://github.com/cosmos/academy-checkers-ui/blob/stargate/test/integration/stored-game.ts" \t "_blank)



Note the forced import of import \_ from "../../environment", to actively inform on the string type (as opposed to string | undefined) and avoid any compilation error.

[#Copy link](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#prepare-your-checkers-chain) Prepare your checkers chain

There is more than one way to run a checkers blockchain. For instance:

* If you came here after going through the rest of the hands-on exercise, you know how to launch a running chain with Ignite.
* If you arrived here and are only focused on learning CosmJS, it is possible to abstract away niceties of the running chain in a minimal package. For this, you need Docker and to create an image:
  1. Get the Dockerfile:



Copy

$ curl -O https://raw.githubusercontent.com/cosmos/b9-checkers-academy-draft/main/Dockerfile-standalone

* 1. Build the image:



Copy

$ docker build . \

-f Dockerfile-standalone \

-t checkersd\_i:standalone

If you have another preferred method, make sure to keep track of the required RPC\_URL accordingly.



If you are curious about how this Dockerfile-standalone was created, head to the [run in production](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/1-run-prod-docker.html) section.

[#Copy link](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#launch-the-tests) Launch the tests

Launch your checkers chain. You can choose your preferred method, as long as it can be accessed at the RPC\_URL you defined earlier. For the purposes of this exercise, you have the choice between three methods:

**Docker standalone**

**Local Ignite**

**Docker Ignite**



Copy

$ docker network create checkers-net

$ docker run --rm -it \

-p 26657:26657 \

--name checkers \

--network checkers-net \

checkersd\_i:standalone start

If your checkers-net network already exists, the first command fails with:



Copy

Error response from daemon: network with name checkers-net already exists

But that is okay.

Copy

$ ignite chain serve

Copy

$ docker network create checkers-net

$ docker run --rm -it \

-v $(pwd):/checkers \

-w /checkers \

-p 26657:26657 \

--name checkers \

--network checkers-net \

checkers\_i \

ignite chain serve

If your checkers-net network already exists, the first command fails with:

Copy

Error response from daemon: network with name checkers-net already exists

But that is okay.

When using Docker, note:

* --name checkers either matches the name you wrote in RPC\_URL, or can be passed as an environment variable to another container to override the value found in .env.
* --network checkers-net, which is reused shortly if you also run your npm tests in Docker. See the paragraph on Docker network, later in this section.

Now, if you run the tests in another shell:

**Local**

**Docker**



Copy

$ npm test

Copy

$ docker run --rm \

-v $(pwd):/client -w /client \

--network checkers-net \

--env RPC\_URL="http://checkers:26657" \

node:18.7-slim \

npm test

This starts the container on the same network as the blockchain container, where checkers resolves to the checkers container. And it also passes RPC\_URL as an override of the value found in .env, typically localhost.

This should return:



Copy

StoredGame

✔ can get game list (39ms)

✔ cannot get non-existent game

SystemInfo

✔ can get system info

3 passing (287ms)

The only combination of running chain / running tests that will not work is if you run Ignite on your local computer and the tests in a container. For this edge case, you should put your host IP address in --env RPC\_URL="http://YOUR-HOST-IP:26657".

[#Copy link](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#a-note-on-docker-networks) A note on Docker networks

You may not have used Docker up to this point. The following paragraphs acquaint you with a Docker *user-defined bridged network*.

If you plan on using Docker Compose at a later stage, having a first taste of such networks is beneficial. Docker Compose can be used to orchestrate and launch separate containers in order to mimic a production setup. In fact, in the [production section](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/1-run-prod-docker.html) of this hands-on exercise you do exactly that. If you think this could eventually be useful, you should go through this section. You may want to redo this section with [Docker (opens new window)↗](https://docs.docker.com/get-docker/).

Earlier you ran the commands:



Copy

$ docker network create checkers-net

$ docker run --rm -it \

-p 26657:26657 \

--name checkers \

--network checkers-net \

--detach \

checkersd\_i:standalone start

This produced the following results:

1. A Docker network was created with the name checkers-net. If containers are started in this network, all ports are mutually accessible.
2. Your container started in it with the resolvable name of checkers.
3. With -p 26657:26657, port 26657 was forwarded to your host computer, on top of being already shared on the checkers-net network.

Then, for tests:

1. When you ran:



Copy

$ npm test

Your tests, running on the **host** computer, accessed the checkers chain from the host computer via the forwarded port 26657. Hence RPC\_URL="http://localhost:26657".

1. When you ran:



Copy

$ docker run --rm \

-v $(pwd):/client -w /client \

--network checkers-net \

--env RPC\_URL="http://checkers:26657" \

node:18.7-slim \

npm test

Your tests, running in a different container, accessed the checkers chain within the checkers-net **Docker network** thanks to the checkers name resolution. Hence RPC\_URL="http://checkers:26657".



In particular, the -p 26657:26657 port forwarding was not necessary. You can confirm that by stopping your chain and starting it again, this time without -p.

Docker networks are explored further in the next section.

When you are done, if you started the chain in Docker you can stop the containers with:



Copy

$ docker stop checkers

$ docker network rm checkers-net

synopsis

To summarize, this section has explored:

* The need to prepare the elements that will eventually allow you to create a GUI and/or server-side scripts for your checkers application.
* How to create the necessary Protobuf objects and clients in TypeScript, the extensions that facilitate the use of these clients, so that CosmJS will understand and be able to interact with your checkers module.
* How to use Docker to define a network to launch separate containers that can communicate, for the purpose of integration testing.

previous

[](https://ida.interchain.io/ida-course/LPs/week-5/)

**[CosmJS Advanced](https://ida.interchain.io/ida-course/LPs/week-5/)**

up next

**[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/2-cosmjs-messages.html)**

[[](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/2-cosmjs-messages.html)](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/2-cosmjs-messages.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

[Compile Protobuf](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#compile-protobuf)

[Prepare integration](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#prepare-integration)

[Integration tests](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#integration-tests)

[Preparation](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#preparation)

[First tests](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#first-tests)

[Prepare your checkers chain](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#prepare-your-checkers-chain)

[Launch the tests](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#launch-the-tests)

[A note on Docker networks](https://ida.interchain.io/hands-on-exercise/3-cosmjs-adv/1-cosmjs-objects.html#a-note-on-docker-networks)

#### **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)