Events

Most entry point functions return a type ofResult .

Within this,Response is a wrapper aroundEvents in the Cosmos SDK.

TheResponse type should be returned as the successful result of a contract entry point (i.e.instantiate orexecute). You can declare it as mutable and add to it in the function body, but a more common pattern is to construct it at the end and return it, if all computation has succeeded. In the examples that follow, it is wrapped byOk as it is being returned as part of a function that is returning theResult type, withResponse representing theRight or success branch.

The exception to this isquery , which will returnStdResultdue to the Cosmos SDK interface.

The source for Response canhelp to understand it better .

The most simple usage ofResponse is as follows:

Ok (Response :: default

( )) This is common ininstantiate functions , where no message is returned to the client.

However, in mostexecute handling cases, aResponse should be returned:

let res =

Response :: new ( ) . add_attribute ("action" ,

"transfer") . add_attribute ("from" , info . sender) . add_attribute ("to" , recipient) . add_attribute ("amount" , amount) ; Ok (res) There's a bit more going on here, so let's unpack it. You can find the sourcehere .

1. A newResponse
2. is created
3. Several key/value pairs are added
4. This is returned wrapped in aResult
5. type usingOk

If you're calling your contract via the command-line interface (CLI) you will see them logged as part of the"raw_log" response, alongside other SDK events.

Instead of just adding attributes,.add_event can be used to add an unwrapped event.

These events can be interacted with by other clients or contracts.Previous Query Next Math