

# Writing A CosmWasm Contract



Ethan Frey

Follow



Jul 10, 2020 · 4 min read



So, you want to write a CosmWasm contract, but don't know where to start? We are here to help sort you out.



Photo by [NESA by Makers](#) on [Unsplash](#)

While we did [write some documentation](#) and [a tutorial back](#) in January to get people started, and updated it with every release, this is not the way most people seem to learn. Now that the [CosmWasm 0.9 release is out](#) and we are very close to a stable API for contract development, it is time to revisit the onboarding experience and help new developers get productive as fast as possible. I do still recommend [reading the architecture overview](#), especially to understand the “actor model” we are using, but after you have built a contract or two.

## Learning By Example

We did start writing [example contracts](#) early on and including some of them as test cases in the [base cosmwasmd repo](#). But these were largely for us to push the limits of the APIs and ensure everything was working properly. Most of the people who look to build on CosmWasm look for analogs to contracts they know from Ethereum: “How do I write an ERC20 token? How do I write UniSwap? Is there a secure multisig contract?”. And with our current stability, we have started writing a collection of them.

[cosmwasm-plus](#) is a new initiative from the CosmWasm team to develop some high-quality examples that you should find useful. As well as producing example contracts, we look to standardize some APIs (like the ERCs did), to make it easy to start combining contracts from different sources. This is still beginning, and no contract there has been audited, but they serve as good examples to learn from and to build upon.

Want to build an ERC777-like contract? Or use those tokens in another contract? We wrote a [CW20 spec](#), which is loosely based on ERC777. And

provide a basic implementation you can extend, as well as an escrow contract that accepts both native tokens and CW20 tokens to show you how to interact with these tokens, almost as easily as native tokens. Beyond providing code you can use as is, or customize, reading through these contracts (and the test cases) will give a good idea of the current best practices in contract development — how to do safe math, store data in buckets, check permissions, and get good test coverage in unit tests.

### Live Workshops

Yeah, I know, I just linked to a whole bunch of test and code samples above... even more daunting than a written tutorial for beginnings. Those are meant to be your virtual textbook, what you turn to for reference. For getting up to speed, we are starting a series of workshops to explain various elements of building CosmWasm contracts. The first workshop goes over the CW20 spec, implementation and escrow contract step-by-step. A 2-hour hands-on experience, examining the code and watching me live-code additions to the contract to get a better feel of the process.

First Workshop, introducing CW20 token contract

We hope to make a series of videos digging deep into various topics. Some topic we are considering are: composing a voting contract and a bonding curve to make our own DAO, calling into native SDK modules, and writing a frontend for your brand new contract with CosmJS. Our next workshop should be late July / early August for the Terra “Hunger Games” Hackathon, but everyone is welcome to join.

Beyond digging in deep to various topics people have asked about, we will also start a slower-paced intro series. For people new to Rust with all kinds of questions on how to get their code to compile (it is a very picky compiler... but does catch a lot of bugs). For people new to blockchain who want to understand better what a CosmWasm contract can and

cannot do (it cannot call any API services, for one). And for people struggling to deploy contracts to a testnet and call them (starting with, which library do you use).

## Tell Us What You Want

We have lots of ideas to present, but most importantly, we want to hear what *you* want to see. Please [sign up on our mailing list](#) if you want to get informed of upcoming workshops, and let us know what topics interest you, so we can cover them. We'd love to see you at our next virtual workshop.

Looking forward to seeing you. In a [workshop](#), on [Github](#) or on [Twitter](#).

Thanks to Misang Ryu.

Cosmwasm

Cosmos Sdk

Tutorial

Workshop



48



WRITTEN BY

**Ethan Frey**

CEO/CTO at cosmwasm.com | multi-chain smart contracts

Follow



**CosmWasm**

The premier smart contract platform for the Cosmos SDK ecosystem. Easy integration to your chains. Live on 3 mainnets and more coming soon!

Follow

## More From Medium

**Second State released SSVM version 0.5.0 for the new generation Ethereum virtual machine Ewasm**

Vivian Hu in Second State



**The SuperBean Project**  
Dr. Superbean



**Elrond—project overview**  
David Filip



**Now you can explain your friends why ETH is worth it's value**

Vamshi Vangapally in HackerNoon.com



**Rocket Pool—Staking Protocol Part 1**

David Rugendyke in Rocket Pool



**How to Use Table Information on EOS Smart Contracts**

ITAM in Coinmonks



**Target Niche for BitNautic Platform 'DIPS'**

BitNautic



**Information Disclosure of EOS Gravity**

EOS Gravity



### Learn more.

Medium is an open platform where 170 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. [Learn more](#)

### Make Medium yours.

Follow the writers, publications, and topics that matter to you, and you'll see them on your homepage and in your inbox. [Explore](#)

### Write a story on Medium.

If you have a story to tell, knowledge to share, or a perspective to offer — welcome home. It's easy and free to post your thinking on any topic. [Start a blog](#)



[About](#) [Write](#) [Help](#) [Legal](#)