Introduction

While developing your smart contract you will want to test that it works as it is supposed to, and moreover, does so securely. In NEAR we have developed tools to help you carry out such tests. Basically, there are two types of tests you can perform:

1. Unit Tests
2. to test methods individually. They are written in the contract's language and are executed locally.
3. Integration Tests
4. to test how your contract behaves in a realistic environment. You can write them in Rust or Typescript, and execute in a local Sandbox or the NEAR testnet.

We recommend all developers to implement both types of tests, since each is suitable to detect different types of errors and make your code intentional. Furthermore, we strongly recommend to first release all projects on testnet, and give users time to try them before releasing them on mainnet.

Setting Up Testing

Testing a smart contract involves using different pieces of software depending on which type of test you are performing. In order to save you from the trouble of setting them up, we recommend you copy the structure from one of ourexample projects . Edit this page Last updatedonApr 10, 2024 bygagdiez Was this page helpful? Yes No Need some help? Chat with us or check ourDev Resources ! Twitter Telegram Discord Zulip

Previous Notes on Serialization Next Unit Testing