Diving deeper

In the last chapter we installed Rust and got up and running with a simple smart contract. The contract has a few issues, however, and isn't as powerful as we'd like it to be. For instance, we can only store one crossword puzzle in the smart contract, the frontend is hardcoded, and we don't offer any incentives to the person who wins.

Let's give the smart contract the ability to store multiple crosswords and offer the winner a prize, paid in NEAR.

Art byr3v.near

In this chapter we'll:

- Allow the contract to store multiple crossword puzzles
- Store the positions of the clues in the contract
- Allow users to log in with a NEAR account
- Give a prize (in NEAR tokens) to the first person to solve the puzzle
- Explore using Rust structs and enums
- more…

A user solves the crossword puzzle, interacts with the blockchain, and gets a prize As we implement the list above, we'll learn key concepts about NEAR:

- Actions
- Full and function-callaccess keys
- NEAR's specializedCollections
- that are generally preferable to, say, Rust's standard HashMap
- The flow of logging in to a decentralized app (dApp)
- more…

Let's jump right in!

Completed project

Here's the final code for this chapter:

https://github.com/near-examples/crossword-tutorial-chapter-2 Edit this page Last updatedonJan 31, 2024 bygagdiez Was this page helpful? Yes No

Previous Add simple frontend Next Store multiple puzzles