

Cairo Crash Course

Components - Exercise 1

In this exercises we will integrate the Ownable component from OpenZeppelin into our contract, and test it



Contract Implementation

You have given an initial contract that has a **value: u32** in the storage, you also have the interface of the contract that you need to implement. The interface contains the following functions:

1. **get_value** - View function that returns the value in the contract storage.
2. **update_value** - External function that updates the value in the contract storage.

In the **working_with_components.cairo** file:

1. Incorporate the **Ownable** component by OpenZeppelin. Follow the steps given in the lecture, and check also the OpenZeppelin [Ownable Component](#) page.
2. Add the ownable component Events and Storage.
3. In the constructor, initialize the ownable component with the account that is being sent as a parameter, also initialize the value in the storage to 100.
4. Load the implementations of the Component (both the External and Internal functions) in the contract.
5. Implement a View function to retrieve the value that is in the contract storage.
6. Implement an External function that can only be called by the owner of the contract. This function should set a value in the contract storage.

In **test_components_2.cairo** file:

You already have implemented function that receives an address as parameter and deploys the contract with that address as the owner.

1. In the first test **test_ownership_component_success**:
 1. Generate an address for alice, and deploy the contract with alice as the owner.
 2. Make sure the initial value in the contract is correct.
 3. Call the **update_value** function with alice as the sender, and check that the value in the contract storage has been updated correctly.
2. Implement the second test **test_ownership_component_fail**:
 1. Generate an address for alice and bob, and deploy the contract with alice as the owner.
 2. Make sure the initial value in the contract is correct.
 3. Call the **update_value** function with bob as the sender, and check that the transaction reverts.

Useful links

[Components](#)