How to write an initializer function

This page explains how to write an initializer function.

To learn more about initializers, readthis .

Initializers are regular functions that set an "initialized" flag (a nullifier) for the contract. A contract can only be initialized once, and contract functions can only be called after the contract has been initialized, much like a constructor. However, if a contract defines no initializers, it can be called at any time. Additionally, you can define as many initializer functions in a contract as you want, both private and public.

Annotate with#[aztec(private)]

and#[aztec(initializer)]

Define your initiaizer like so:

[aztec(private)]

[aztec(initializer)]

fn

constructor ( ) { // function logic here }

Initializer with logic

Initializers are commonly used to set an admin, such as this example:

constructor

[aztec(public)]

[aztec(initializer)]

fn

constructor (admin :

AztecAddress , name :

str < 31

, symbol :

str < 31

, decimals :

u8)

{ assert (! admin . is_zero ( ) ,

"invalid admin") ; storage . admin . write (admin) ; storage . minters . at (admin) . write (true) ; storage . name . initialize (FieldCompressedString :: from_string (name)) ; storage . symbol . initialize (FieldCompressedString :: from_string (symbol)) ; storage . decimals . initialize (decimals) ; } Source code: noir-projects/noir-contracts/contracts/token_contract/src/main.nr#L54-L67 Here, the initializer is calling a public function. It can also call a private function. Learn more about calling functions from functionshere .

To see constructors in action, check out theAztec.nr getting started guide . Edit this page

Previous How to call functions from other functionsNext How to define compute_note_hash_and_nullifier