

Enseignant(s)

HILLARD Franck

Email(s)

fhillard@myges.fr

Tezos_ligo_vesting_contract_2024

1 Matières, formations et groupes

Matière liée au projet :

Formations : -

Nombre d'étudiant
par groupe :

1

Règles de constitution des groupes: **Imposé**

Charge de travail
estimée par étudiant : **25,00 h**

2 Sujet(s) du projet

Type de sujet : **Imposé**

Contrat de vesting avec fonds sequestrés

Vesting contract (locked tokens)

use cases

The Vesting contract can be used task related to fund distribution (ICO, corporate actions).

What is asked

Create a smart contract (called Vesting) that distributes funds to beneficiaries on a period of time. Funds are implemented as a FA2 token (TZIP-12).

Funds are first frozen during a freeze period (i.e. funds cannot be claimed). Then funds are available (i.e. claimable) on time basis. During the vesting period, funds are claimable proportionnaly to the vesting period duration. At the end of the vesting period, 100% of funds are claimable.

The administrator of the Vesting contract is the user who deployed the Vesting contract.

The administrator can call the `Start` entrypoint which will trigger the beginning of the freeze period and the lock of funds (i.e. fund transfer from administrator to the Vesting contract) . Once the Vesting contract is started, the beneficiaries cannot be changed, and vesting start time and end time cannot be changed.

The beneficiaries are specified at the creation of the contract, with their corresponding promised amounts of token.

An entrypoint `UpdateBeneficiary` must be provided to modify the beneficiaries. This entrypoint must be callable only by the administrator if the Vesting contract is not started yet.

The vesting duration, and freeze period duration are specified at the creation of the contract.

The FA2 token (address and token_id) that is used to represent funds must be specified at the creation of the contract.

Available funds can be claimed by a beneficiary. The `claim` entrypoint transfers available amount of tokens (and which has not been claimed yet) to the beneficiary.

A `kill` entrypoint callable only by the administrator must be implemented to be able to retrieve funds, and pay beneficiaries (on time elapsed basis) and to clean the storage.

Obviously,

- a non-beneficiary cannot claim and receive funds
- a beneficiary cannot claim and receive more funds than promised

Hints

You will need the "FA2" token implementation from Ligolang registry.

The `claim` entrypoint transfers available funds to the beneficiary (caller) by invoking the `Transfer` entrypoint of the FA2 token.

What is expected

A github repository (with access to github user frankhillard) which contains at least the following points :

- a complete README (what this repository does , how to use it, limitations, ..)
- a `src` directory with smart contract sources
- a `test` directory with exhaustive tests in ligo (tests must check success and failure of entrypoints)
- a `script` directory with a deployment script , and one or many integration tests (consisting on TypeScript scripts that interact with a deployed contract and ensures that the contract works correctly.
- a makefile (or something equivalent) will be appreciated
- a .gitignore file
- a ligo.json file

3 Détails du projet

Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)

Développement avec un langage fonctionnel (LIGO) et dans un contexte blockchain (TEZOS).

Descriptif détaillé

Implémentation d'un contrat de Vesting permettant la distribution de fonds (crypto-monnaie) à des utilisateurs sur une période de temps.

Ouvrages de référence (livres, articles, revues, sites web...)

Outils informatiques à installer

Pour les contrats (et tests): compilateur "ligo" (syntaxe "cameligo"); pour les tests d'intégration (et de déploiement): librairie "taquito".

4 Livrables et étapes de suivi

5 Soutenance

Durée de présentation par groupe : **15 min** Audience : **A huis clos**

Type de présentation : **Démonstration - Autre**

Précisions : **video conférence**