**Tasks**

**1) ERC-20 token**

**name**: ILTS

**total supply**: 100000

**decimals:** 2

**contract:**

<https://testnet.snowtrace.io/address/0xB87232D05f0F24aBE84ee5c17f59f190d773099B>

**2) Vesting contract**

<https://testnet.snowtrace.io/address/0x95374C79Fd82A4F0C6750612c7e38296e88F0b2e>

**Methods:**

- create allocation for address

*dynamic allocation per address which can be presented like set of periods (10 days = 25%, 30 days = 35%, 50 days = 40% and any amount of periods and percentages)*

- change allocation for address (change allocation amount)

- claim token

- claim allowance (allowed to be claimed per address)

**Vesting Contract Implementation**

**Data structure**

Text

Description automatically generated

**Create allocation for address**

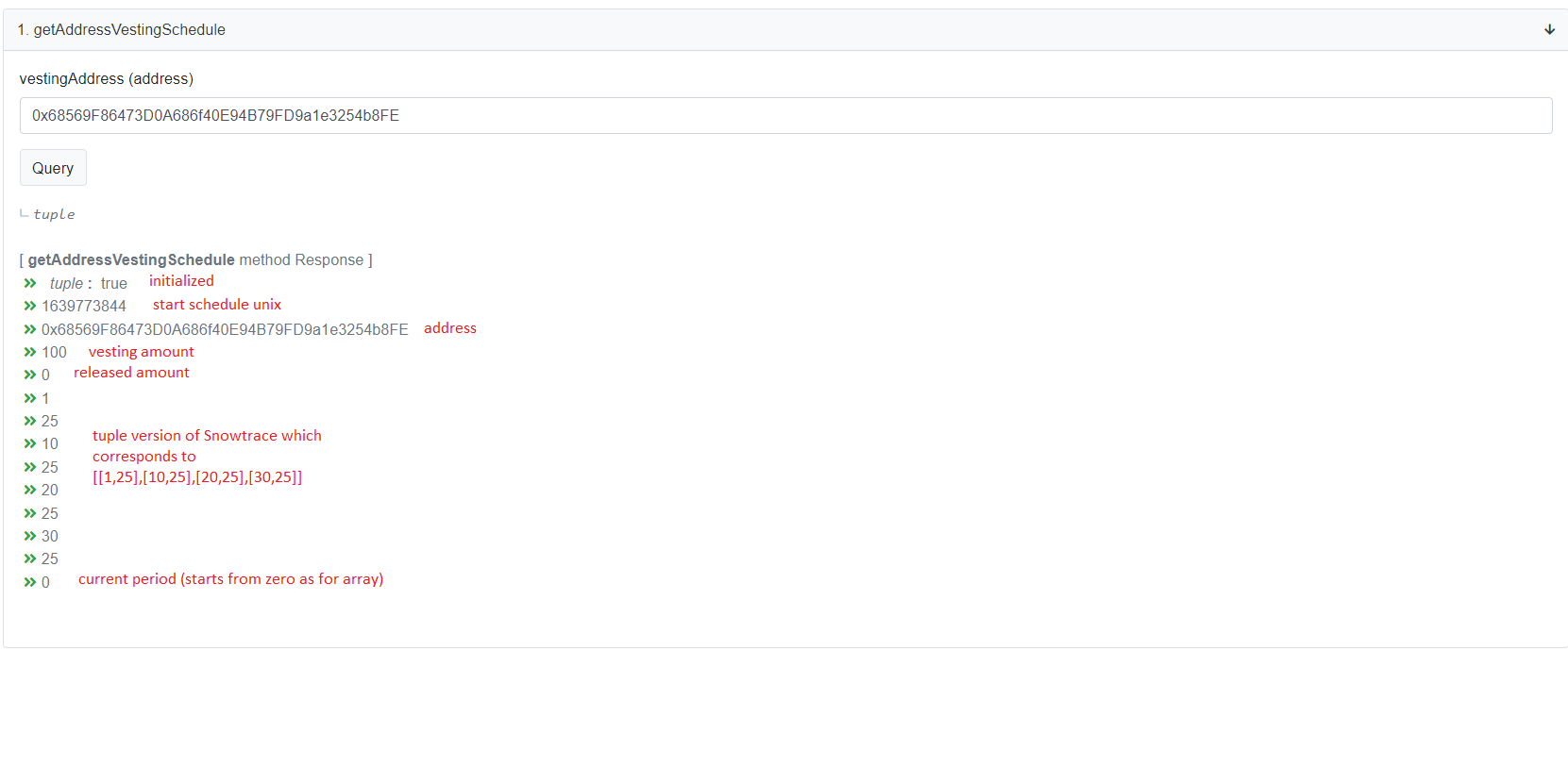
Text

Description automatically generated

**Parameters**:  
- vesting address  
- vesting amount  
- start in unix  
- periods tuple  
  
**Requirements**:  
- only owner can call the function  
- schedule/allocation per address can be created only once  
- periods tuple should be following pattern   
  
[[10,100]]  
[[10,25],[20,25],[30,25],[40,25]]  
  
where first parameter its days from start vesting schedule to open claim wave (each next period should be greater than previous)  
where second parameter its percentage of total tokens amount to be allowed to claim (total for all periods should be 100%)

**Usage**:  
Graphical user interface, text, application, email

Description automatically generated  
Start unix is taken from there:  
https://www.unixtimestamp.com/  
Graphical user interface, text, application, email

Description automatically generated  
  
**Check schedule**:  


**Get amount of tokens that can be claimed for the address right now**:  
*connected to example above*  
A picture containing graphical user interface

Description automatically generated

**Claim tokens:**Text

Description automatically generated **Parameters**:  
- vesting address  
- claim amount   
  
**Requirements**:  
- only if vesting schedule for address exists  
- only owner or address related to schedule can request claim  
- amount of tokens requested should be not more than allowed in this period  
- requester should act when claim is allowed  
  
**Usage:**   
Before usage we should mint all ILTS tokens to vesting smart contract (lockup)  
ILTS smart contract mint usage:  
A picture containing background pattern

Description automatically generated

Then we can call claim tokens for address  
Application

Description automatically generated with medium confidence  
Balance after claim:  
Graphical user interface, text, application

Description automatically generated  
  
**Change allocation:**Table

Description automatically generated with low confidenceAllows to change amount of tokens to be claimed into specific schedule per address.