Executor Contract

Introduction

The Executor contract allows pullPayment contracts to execute the pullPayment. Executor allows only registered pullPayment contracts to transfer the tokens from customer to merchant using the executor() method.

The Executor contract gets the payment tokens from the customer, swaps the tokens to settlement tokens with the help of the uniswap router and transfers the settlement tokens to the merchant.

Contract version

pragma solidity 0.8.0

Contract constructor

In this contract the initialize() method is used in place of the constructor to allow the contract to be upgradable via proxy.

Method Name:

- initialize

Method Usage:

- The initialize method is used to initialize the Executor contract, set the registry and pullPayment registry address.

Method Signature:

- function initialize(address registryAddress, address ppRegistryAddress)
  external;

Method Parameters:

1. **registryAddress:**
   - indicates the address of the registry contract which keeps the record of all the contract addresses in the ecosystem.
2. **ppRegistryAddress:**
   - Indicates the address of pullPayment registry which keeps the record of pullPayment contracts only.

Contract Methods

execute()

Method name:

- execute

Method detail:

- This method executes the pullPayment of given subscription id of billing model for given pullPayment contract type.
- Internally this method first gets the address of the pullPayment contract from the pullPayment registry contract and calls the executePullPayment() method of that pullPayment contract.
- The executePullPayment() method of pullPayment contract further calls the execute method of Executor contract by passing the sender, receiver, settlement token address, payment token address and payment amount.
- Anyone can call this method by providing the valid parameters.

Method signature:

- function execute(string calldata _bmType, uint256 _subscriptionId)
  external
  override
  returns (uint256 pullPaymentID)

Method parameters:

1. **_bmType:**
   - Indicates the name of the pullPayment contract whose pullPayment is to execute. bmType is specified in the bytes32 format.
2. **_subscriptionId:**
   - Indicates the subscription id whose pullPayment is to execute.

Returned data:

- This method returns the unique pullPayment id which is just executed.

execute()

Method name:

- execute

Method detail:

- The execute method executes the pullPayment for the pullPayment contracts.
- Only the registered and granted pullPayment contracts can call this method.
- Internally this method first gets the amount of tokens from the customer, converts the payment tokens to settlement tokens if needed and then transfers the converted tokens to the merchant.
- It uses the uniswapV2 router for swapping the tokens.

Method signature:

- ```
  function execute(
      address settlementToken,
      address paymentToken,
      address from,
      address to,
      uint256 amount
  ) external override onlyGrantedExecutors(msg.sender) returns (bool)
  ```

Method parameters:

- **settlementToken:**
  - Indicates the token address in which the merchant wants to get paid in.
- **paymentToken:**
  - Indicates the payment token address in which the customer wants to pay for the subscription.
- **from:**
  - Indicates the customer address from which tokens are transferred to the merchant.
- **to:**
  - Indicates the merchant address.
- **amount:**
  - Indicates the amount of tokens that merchants should receive.

Returned data:

- This method returns true if the pullPayment executes successfully otherwise it returns false.
  ```

Flow of Execution:

1. First we set the main registry and pullPayment registry addresses at the time of contract initialization.
2. The pullPayment contract who wants to execute their pullPayments for the subscription has to be registered on the pullPayment registry contract.
3. Anyone can execute the pullPayment for a given subscription using the external execute method, which requires the pullPayment contract name and the subscription id.
4. The external execute method gets the address of the pullPayment contract from the pullPayment Registry and creates the interface for the pullPayment contract.
5. Further, it calls the executePullPayment() method of the pullPayment contract.
6. The pullPayment contract further calls the execute method of the Executor contract to pull the payment from the customer and transfer it to the merchant.