Registry Contract

Introduction

This Registry contract is a central contract which keeps the record of all the contracts in the ecosystem. It stores the contract name associated with the contract addresses. Anyone can get the addresses of any contract

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 Registry contract and transfers the ownership of registry to the admin who deploys the registry.

Method Signature:

- function initialize() external;

Method Parameters:

- No parameters required.

Contract Methods

setAddressFor()

Method name:

- setAddressFor

Method detail:

- This method registers the given contract address with the given name.
- Only the registry owner can register the contract.

Method signature:

- function setAddressFor(string calldata identifier, address addr) external override

Method parameters:

- **_identifier:**
    - Indicates the name of the contract in the ecosystem that we want to register.
    - Identifier is specified in the readable string format
- **_addr:**
    - Indicates the address of the contract.

Returned data:

- This method does not return any data.

getAddressForOrDie()

Method name:

- getAddressForOrDie

Method detail:

- This method retrieves the contract`s address associated with the given identifier hash.
- If it does not find the contract`s address for a given identifier hash, it throws an error.

Method signature:

-
  ```
  function getAddressForOrDie(bytes32 _identifierHash)
      external
      override
      view
      returns (address)
  ```

Method parameters:

- **_identifierHash:**
    - Indicates the contract name in bytes32 format.

Returned data:

- This method returns the address of the contract associated with the identifier.

getAddressFor()

Method name:

- getAddressFor

Method detail:

- This method retrieves the contract address for the given identifier hash.
- If it does not get the address for the given identifier, it returns the default zero address.

Method signature:

-
  ```
  function getAddressFor(bytes32 _identifierHash)
      external
      override
      view
      returns (address)
  ```

Method parameters:

- **_identifierHash:**
  - Indicates the pullPayment contract name in bytes32 format.

Returned data:

- This method returns the pullPayment contract address if it exists or it returns the zero-address.

getAddressForStringOrDie()

Method name:

- getAddressForStringOrDie

Method detail:

- This method returns the contract address for the given contract name.
- It throws an error if it does not find the address for the given contract name.

Method signature:

-
  ```
  function getAddressForStringOrDie(string calldata _identifier)
      external
      view
      override
      returns (address)
  ```

Method parameters:

- **_identifier:**
  - Indicates the pullPayment contract name in string format.

Returned data:

- This method returns the pullPayment contract address if it exists.

getAddressForString()

Method name:

- getPPAddressForString

Method detail:

- This method retrieves the address of the pullPayment contract for a given contract name.
- It returns the zero-address if it does not find the contract address associated with the given contract name.

Method signature:

- function getAddressForString(string calldata _identifier)
  external
  view
  override
  returns (address)

Method parameters:

- **_identifier:**
  - Indicates the name of the pullPayment contract in string format.

Returned data:

- This method returns the address of the pullPayment contract.

isOneOf()

Method name:

- isOneOf

Method detail:

- This method tells you whether the given address is registered in the registry or not.
- We need to provide the list of contract identifier hashes and the sender address whose membership to be verified.

Method Signature:

-
  ```
  function isOneOf(bytes32[] calldata identifierHashes, address sender)
    external
    override
    view
    returns (bool)
  ```

Method parameters:

- **identifierHashes:**
  - Indicates the list of contract identifier hashes.
- **sender:**
  - Indicates the contract address.

Returned data:

- This method returns true if the given address is one of the registered contract otherwise it returns false.

Events

RegistryUpdated

Event Name:

- RegistryUpdated

Event detail:

- This event is emitted when a new contract is registered using the **setAddressFor()** method.

Event signature:

-
  ```
  event RegistryUpdated(
        string identifier,
        bytes32 indexed identifierHash,
        address indexed addr
     );
  ```

Event data:

- **identifier:**
  - Indicates the name of the contract.
- **identifierHash:**
  - Indicates the contract name in bytes32 format.
- **Addr:**
  - Indicates the address of the registered contract.

Flow of Execution:

1. When we add the new contract in the ecosystem which, that contract must be registered on the main Registry contract.
2. Owner registers the new contract on the registry.
3. Get addresses of registered contracts either providing the contract name or hash of the contract name.