

PortfolioManager

Jan Kwiatkowski*swapExactInputSingle*

Menadżer Portfolia

inteligentny kontrakt pozwalający na składanie zleceń na dane tokeny

Methods

DEFAULT_ADMIN_ROLE

`function` DEFAULT_ADMIN_ROLE() `external view` returns (bytes32)

Returns

Name	Type	Description
_0	bytes32	undefined

KEEPER_ROLE

`function` KEEPER_ROLE() `external view` returns (bytes32)

hasz roli KEEPER_ROLE

Returns

Name	Type	Description
_0	bytes32	undefined

addOrder

`function` addOrder(address _asset, enum PortfolioManager.OrderType _orderType, int256 _pr

funkcja pozwalająca na dodawanie zleceń

Parameters

Name	Type	Description
_asset	address	adres tokenu którego dotyczy zlecenie
_orderType	enum PortfolioManager.OrderType	typ zlecenia kupno/sprzedaż
_price	int256	cena realizacji zlecenia
_amount	uint256	ilość tokenów które chcemy kupić/sprzedać

executeOrders

```
function executeOrders(uint256[] _orders) external nonpayable
```

funkcja wykonująca zlecenia przekazane w tabeli

Parameters

Name	Type	Description
_orders	uint256[]	zlecenia kwalifikujące się do wykonania

getEligibleOrders

```
function getEligibleOrders() external view returns (uint256[])
```

zwraca zlecenia które kwalifikują się do wykonania

Returns

Name	Type	Description
_0	uint256[]	undefined

getRoleAdmin

```
function getRoleAdmin(bytes32 role) external view returns (bytes32)
```

Returns the admin role that controls `role` . See `{grantRole}` and `{revokeRole}`. To change a role's admin, use `{_setRoleAdmin}`.

Parameters

Name	Type	Description
role	bytes32	undefined

Returns

Name	Type	Description
_0	bytes32	undefined

grantRole

```
function grantRole(bytes32 role, address account) external nonpayable
```

Grants `role` to `account` . If `account` had not been already granted `role` , emits a `{RoleGranted}` event. Requirements: - the caller must have `role` 's admin role. May emit a `{RoleGranted}` event.

Parameters

Name	Type	Description
role	bytes32	undefined
account	address	undefined

hasRole

```
function hasRole(bytes32 role, address account) external view returns (bool)
```

Returns `true` if `account` has been granted `role` .

Parameters

Name	Type	Description
role	bytes32	undefined

Name	Type	Description
account	address	undefined

Returns

Name	Type	Description
_0	bool	undefined

poolFee

```
function poolFee() external view returns (uint24)
```

Returns

Name	Type	Description
_0	uint24	undefined

priceConsumer

```
function priceConsumer() external view returns (contract PriceConsumerV3)
```

kontrakt dostarczający dane o cenach tokenów

Returns

Name	Type	Description
_0	contract PriceConsumerV3	undefined

renounceRole

```
function renounceRole(bytes32 role, address account) external nonpayable
```

Revokes `role` from the calling account. Roles are often managed via `{grantRole}` and `{revokeRole}`: this function's purpose is to provide a mechanism for accounts to lose their privileges if they are compromised (such as when a trusted device is misplaced). If the calling account had been revoked

role , emits a *{RoleRevoked}* event. Requirements: - the caller must be *account* . May emit a *{RoleRevoked}* event.

Parameters

Name	Type	Description
role	bytes32	undefined
account	address	undefined

revokeRole

```
function revokeRole(bytes32 role, address account) external nonpayable
```

Revokes role from account . If account had been granted role , emits a {RoleRevoked} event. Requirements: - the caller must have role 's admin role. May emit a {RoleRevoked} event.

Parameters

Name	Type	Description
role	bytes32	undefined
account	address	undefined

supportsInterface

```
function supportsInterface(bytes4 interfaceId) external view returns (bool)
```

See {IERC165-supportsInterface}.

Parameters

Name	Type	Description
interfaceId	bytes4	undefined

Returns

Name	Type	Description
_0	bool	undefined

swapExactInputSingle

```
function swapExactInputSingle(uint256 amountIn, address tokenIn, address tokenOut, address recipient)
```

funkcja wymieniająca jedne tokeny na drugie

Parameters

Name	Type	Description
amountIn	uint256	dokładna ilość tokenu wchodzącego, mającego być zamieniony na token wychodzący
tokenIn	address	token który sprzedajemy
tokenOut	address	token który kupujemy
recipient	address	undefined

Returns

Name	Type	Description
amountOut	uint256	undefined

swapRouter

```
function swapRouter() external view returns (contract ISwapRouter)
```

Returns

Name	Type	Description
_0	contract ISwapRouter	undefined

weth

```
function weth() external view returns (address)
```

adres tokenu Wrapped Ethereum

Returns

Name	Type	Description
_0	address	undefined

Events

OrderAdded

```
event OrderAdded(address indexed _asset, int256 price)
```

event dla dodania zlecenia

Parameters

Name	Type	Description
_asset indexed	address	undefined
price	int256	undefined

RoleAdminChanged

```
event RoleAdminChanged(bytes32 indexed role, bytes32 indexed previousAdminRole, bytes32
```

Parameters

Name	Type	Description
role indexed	bytes32	undefined
previousAdminRole indexed	bytes32	undefined

Name	Type	Description
newAdminRole indexed	bytes32	undefined

RoleGranted

`event RoleGranted(bytes32 indexed role, address indexed account, address indexed sender)`

Parameters

Name	Type	Description
role indexed	bytes32	undefined
account indexed	address	undefined
sender indexed	address	undefined

RoleRevoked

`event RoleRevoked(bytes32 indexed role, address indexed account, address indexed sender)`

Parameters

Name	Type	Description
role indexed	bytes32	undefined
account indexed	address	undefined
sender indexed	address	undefined