

概要

4626标准允许为代表单个底层 ERC-20 代币份额的代币化Vault实施标准 API。该标准是 ERC-20 代币的扩展，它提供了存取代币和读取余额的基本功能。

Vault是什么？

Vault是一个旨在通过锁定代币来累积收益的智能合约。Vaults 有一个指定的策略，旨在通过资本投资、自动复利和投资组合再平衡来产生回报。

+ :: **Example:**

你将 \$DAI 存入一个发行有收益的代币 \$vDAI 的Vault。这种带有收益的代币通过应用创收策略随着时间的推移而积累价值。带有收益的代币代表Vault内资产池的部分所有权。如果Vault池的价值增长，那么你的代币的价值也会增加，因为它代表了你在池中的份额。

动机

代币化Vault缺乏标准化合约，导致实施方法难以统一，例子包括借贷市场、聚合器和具有内在利息的代币。这使得需要符合许多标准的协议在聚合器或插件层难以集成，并迫使每个协议实现自己的适配器，这些适配器容易出错并浪费开发资源。代币化Vault的标准将降低收益Vault的集成工作量，同时创建更加一致和强大的通证范式。

ERC-20标准

Methods:

- Token名字

复制代码

```
1 function name() public view returns (string)
```

javascript >

- Token代码

复制代码

```
1 function symbol() public view returns (string)
```

javascript >

- 小数位数

复制代码

```
1 function decimals() public view returns (uint8)
```

javascript >

- 总供给

复制代码

```
1 function totalSupply() public view returns (uint256)
```

javascript >

- 查询余额

复制代码

```
1 function balanceOf(address _owner) public view returns (uint256 balance)
```

javascript >

- 转账

复制代码

```
1 function transfer(address _to, uint256 _value) public returns (bool success)
```

javascript >

- 授予权限

复制代码

```
1 function approve(address _spender, uint256 _value) public returns (bool success)
```

javascript >

- 授权余额

复制代码

```
1 function allowance(address _owner, address _spender) public view returns (uint256 remaining)
```

javascript >

<https://etherscan.io/token/0xbb0e17ef65f82ab018d8edd776e8dd940327b28b#readContract>

<https://etherscan.io/address/0xbb0e17ef65f82ab018d8edd776e8dd940327b28b#code>

ERC-4626标准

Methods:

- totalAssets
- convertToShares
- convertToAssets
- maxDeposit / previewDeposit / Deposit
- maxMint / previewMint / Mint
- Withdraw
- Redeem

优点

- 大幅增加不同协议的互操作性
- 大大降低了开发的复杂性和工作量
- 提高智能合约的安全性和可审计性
- 标准化将开启前所未有的创新