**TokenManager-MappingToken-**

**Oracle-**

**Robot-**

**Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | 2020-6-10 | jishiwu | Initial draft |
| 0.2 | 2020-6-22 | jishiwu |  |
| 0.3 | 2020-7-13 | jishiwu |  |
| 0.4 | 2020-7-31 | jishiwu | Add indexed for events add admin for oracle |
| 0.5 | 2020-8-06 | jishiwu | Merge map,  change address to bytes changeTokenOwner  acceptOwnership |

# TokenManager

# 

# 职责

# 管理从别的链映射到本链的虚拟token

# 合约接口设计

合约结构定义：

struct AncestorInfo {

bytes account;

string name;

string symbol;

uint8 decimals;

uint chainID;

}

struct TokenPairInfo {

uint fromChainID; /// index; e.g. eth=60, etc=61, wan=5718350

bytes fromAccount; /// from address

uint toChainID; /// same as before

~~address tokenAddress;~~ /// to token address

bytes toAccount; // tokenAddress → toAccount

~~bool isDelete;~~ /// whether been deleted

}

合约变量:

uint public totalTokenPairs = 0;

mapping(address => bool) public mapAdmin;

~~mapping(uint => AncestorInfo) public mapAncestorInfo;~~

mapping(uint => TokenPairInfo) public mapTokenPairInfo;

mapping(uint => uint) public mapTokenPairIndex;

合约事件

event AddToken(address tokenAddress, string name, string symbol, uint8 decimals);

event AddTokenPair(uint indexed id, uint fromChainID, bytes fromAccount, uint toChainID, **address toAccount**);

event UpdateTokenPair(uint indexed id, bytes ancestorAccount, string ancestorName, string ancestorSymbol, uint8 ancestorDecimals,

uint ancestorChainID,uint fromChainID, bytes fromAccount, uint toChainID, **bytes toAccount**, **~~bool isValid~~**);

event RemoveTokenPair(uint indexed id);

~~event MintToken(uint~~ **~~indexed~~** ~~id, address to, uint value);~~

~~event BurnToken(uint~~ **~~indexed~~** ~~id, uint value);~~

event AddAdmin(address admin);

event RemoveAdmin(address admin);

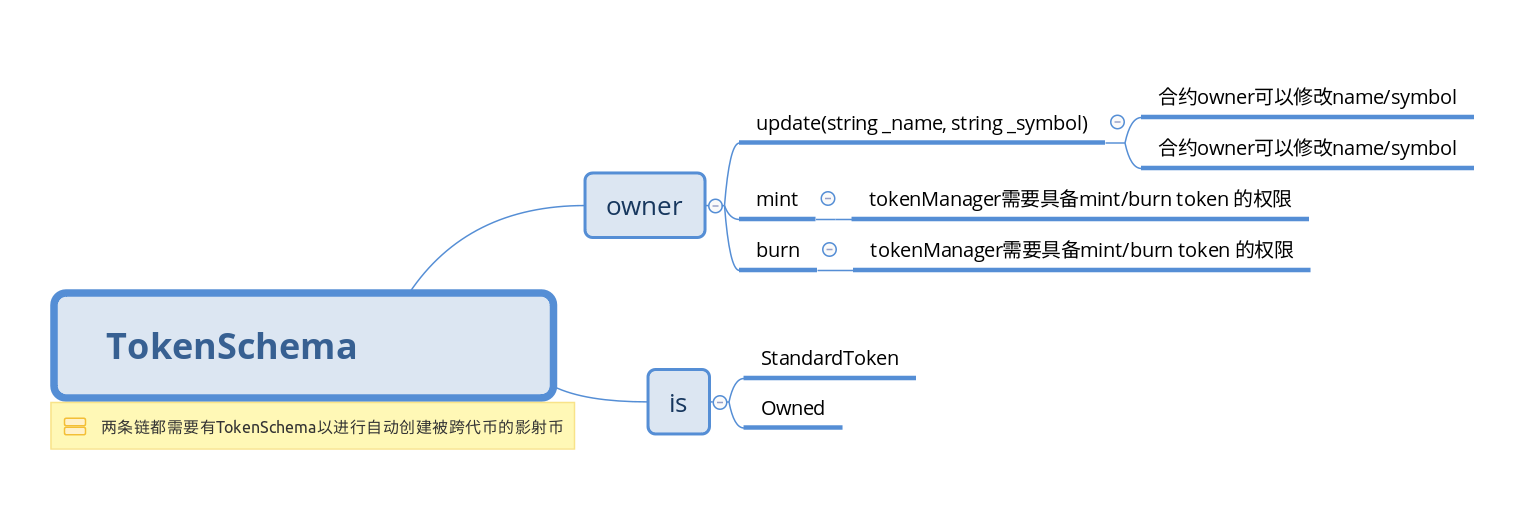
event UpdateToken(**address tokenAddress**, string name, string symbol);

|  |  |  |  |
| --- | --- | --- | --- |
|  | **权限** | **函数名** | **功能** |
| 1 | owner | function addTokenPair(  uint id,  AncestorInfo aInfo,  uint fromChainID,  bytes fromAccount,  uint toChainID,  ~~address tokenAddress~~  bytes toAccount  ) | 创建一个新token映射 event UpdateAncestorInfo(id, ancestorAccount, ancestorName, ancestorSymbol, ancestorChainID);  id从1算起 |
| 2 | owner | function addToken(  string name,  string symbol,  uint8 decimals  ) | 需要解析事件，获取新token的地址  event TokenAdd(tokenAddress, name, symbol, decimals) |
| 4 | owner | function updateTokenPair(  uint id,  AncestorInfo aInfo,  uint fromChainID,  bytes fromAccount,  uint toChainID,  ~~address tokenAddress~~  bytes toAccount  ) | 更新token映射信息 event UpdateTokenPair(id, fromChainID, fromAccount, toChainID, tokenAddress) |
| 5 | owner | function removeTokenPair(  uint id  ) | 删除token映射  event RemoveTokenPair(id) |
| 6 | **admin** | function mintToken(  uint id,  address to,  uint value  ) | 给to，铸造value个类型为id的token event MintToken(uint id, address to, uint value) |
| 7 | **admin** | function burnToken(  uint id,  uint value  ) | event BurnToken(uint id, uint value) |
| 10 | owner | function addAdmin(  address admin  ) | 设置地址为admin权限 |
| 11 | owner | function removeAdmin(  address admin  ) | 取消地址的admin权限 |
| 15 | owner | function updateToken(  ~~uint id,~~  **address tokenAddress,**  string name,  string symbol  ) | 更新token，修改name,symbol |
| 12 | **external** | function getTokenPairInfo(  uint id  ) external view returns (  uint fromChainID,  bytes fromAccount,  uint toChainID,  ~~address tokenAddress,~~  bytes toAccount,  ~~bool isValid~~  ) | 获取跨链币信息 |
| 13 | **external** | function getTokenInfo(uint id) external view returns (  address addr,  string name,  string symbol,  uint8 decimals  } | 获取编号为id的token信息 |
| 14 | **external** | function getAncestorInfo(uint id) external view returns (  bytes account,  string name,  string symbol,  uint8 decimals,  uint chainId  ) | 获取祖先token信息 |
| 16 | **external** | function getTokenPairs()  external view returns(  uint[] id,  uint[] fromChainID,  bytes[] fromAccount,  uint[] toChainID,  ~~address[] tokenAddress,~~  bytes toAccount,  string[] ancestorSymbol,  uint8[] ancestorDecimals) | 获取所有的tokenPair信息 |
| 17 | **external** | function getTokenPairsByChainID(  uint chainID1,  uint chainID2  ) external view returns (  uint[] id,  uint[] fromChainID,  bytes[] fromAccount,  uint[] toChainID,  ~~address[] tokenAddress,~~  bytes toAccount,  string[] ancestorSymbol,  uint8[] ancestorDecimals  ) | 获取chainID1,chainID2之间的跨链token |
| **18** | **owner** | **function changeTokenOwner(**  **address tokenAddress,**  **address \_newOwner**  **) external onlyOwner** | 转移token的owner, 需要调用token的acceptOwnership |
| **19** | **owner** | **function acceptTokenOwnership(  address tokenAddress ) external** | **接受ownership的转移请求** |

**Mapping****Token**

1. **职责**

此为token模板，包含erc20的全部功能，且多了mint和burn接口，此模板的对象由tokenManager管理。他是其它链上的加密货币，在本地链上的映射。添加originChain字段，指示原始加密货币属于哪条链，例如1 – wan、2 – eth、3 – etc



1. **合约**

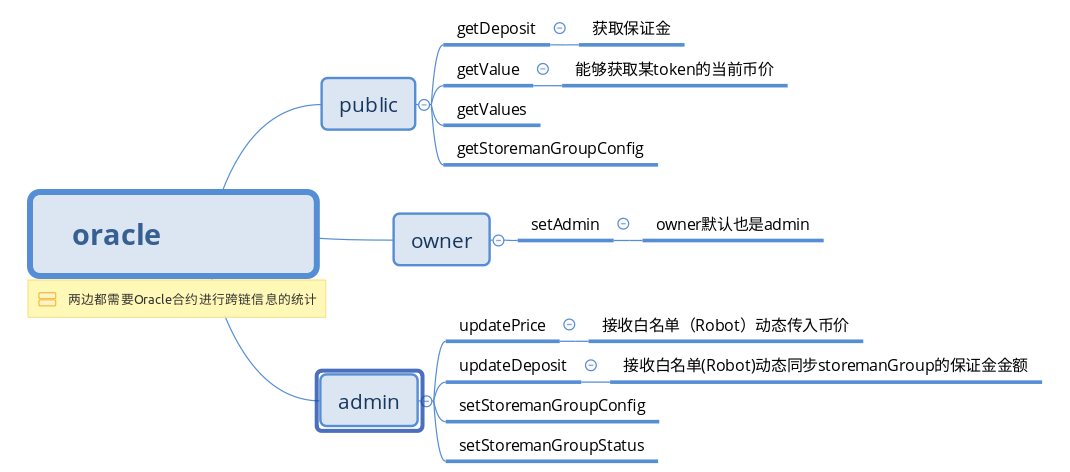
|  |  |  |  |
| --- | --- | --- | --- |
|  | **权限** | **函数名** | **功能** |
| 1 | owner | mint(address account, uint value) | 为某账户铸造加密币 |
| 2 | owner | burn(address account, uint value) | 为某账户销毁加密币 |
| 4 | owner | update(string \_name, string \_symbol) | 更新name,symbol |

**Oracle**

**1.职责**

更新原始加密货币的usd价格，提供原始加密货币的usd价格,监控storeman剩余押金

**2.合约**



|  |  |  |  |
| --- | --- | --- | --- |
|  | **权限** | **函数名** | **功能** |
| 1 | admin | updatePrice(bytes32[] keys, uint[] prices) | 更新价格 |
| 2 | public | getValue(bytes32 key) | 获取某加币货币的价格 |
| 3 | public | getValues(bytes32[] key) |  |
| 4 | admin | updateDeposit(bytes32 smgID, uint amount) | 更新保证金 |
| 5 | public | getDeposit(bytes32 smgID) | 获取保证金额 |
| 6 | admin | setStoremanGroupStatus(bytes32 id, uint8 status) | 设置storeman状态 |
| 7 | admin | SetStoremanGroupConfig(  bytes32 id,  uint8 status,  uint deposit,  uint[2] chain,  uint[2] curve,  bytes gpk1,  bytes gpk2,  uint startTime,  uint endTime  ) | 设置storemanGroup配置 |
| 8 | public | getStoremanGroupConfig(bytes32 id) external view returns(  bytes32 groupId,  uint8 status,  uint deposit,  uint chain1,  uint chain2,  uint curve1,  uint curve2,  bytes gpk1,  bytes gpk2,  uint startTime,  uint endTime) | 获取storemanGroup配置信息 |
| 9 | owner | setAdmin(address addr) external onlyOwner | 设置admin |

**Robot**

**1.职责**

获取第三方市场提供的原始加密货币价格，并更新到oracle中

**2.功能**

|  |  |  |
| --- | --- | --- |
|  | **函数名** | **功能** |
| 1 | updateDeposit(bytes32 smgID, int[] prices) | 向oracle更新保证金金额 |
| 2 | updatePrice(bytes[] keys, uint[] prices) 满足百分比变动的token更新 | 向oracle更新价格 |
| 3 | syncConfigToOtherChain | 同步smg配置到其他链 |