### ViolasToken 支持多币种的单一move合约说明文档

1. **名词解释**
   1. Supervisor：超级管理员

负责发布module到violas链上

负责调用create\_token函数来发行新的币种

* 1. Owner: 币的铸币人

负责调用mint来铸币

* 1. tokenidx: 币种编号

supervisor调用create\_token会在event种产生币种编号，从0开始，依次递增。

* 1. orderidx: 币币交易挂单编号

用户调用make\_order会在event种产生挂单编号，类型为u64。

1. **函数介绍**
   1. publish(userdata: vector<u8>)

用户使用合约之前必须首先调用publish函数进行合约的注册

Userdata: 用户私有数据，不可修改。

* 1. create\_token(owner: address, tokendata: vector<u8>)

发行新的币种， 只能有supervisor调用

Owner：发行的币种的铸币人

Tokendata： 币种的私有数据，不可修改

**event输出：tokenidx 当前发行的币种编号**

* 1. mint(tokenidx: u64, payee: address, amount: u64, data: vector<u8>)

铸币，只有owner才能铸币

Tokenidx：币种编号

Payee： 币接收人

Amount：币的数量

Data： 用户自定义数据，会写入event中。

* 1. transfer(tokenidx: u64, payee: address, amount: u64, data: vector<u8>)

转账

Tokenidx： 币种编号

Payee： 币接收人

Amount：币的数量

Data： 用户自定义数据，会写入event中。

* 1. make\_order(idxa: u64, amounta: u64, idxb: u64, amountb: u64, data: vector<u8>)

币币交易的maker的挂单操作，**注意：输出的挂单编号会写入event中**。

Idxa：用户卖出的币种编号

Amounta：用户卖出的币种数量

Idxb：用户买入的币种编号

Amountb：用户买入的币种数量

Data： 用户自定义数据，会写入event中。

* 1. cancel\_order(orderidx: u64, idxa: u64, amounta: u64, idxb: u64, amountb: u64, data: vector<u8>)

币币交易的maker的撤单操作，**注意：idxa，amounta，idxb，amountb必须和挂单操作一样。**

Orderidx：挂单编号

Idxa：用户卖出的币种编号

Amounta：用户卖出的币种数量

Idxb：用户买入的币种编号

Amountb：用户买入的币种数量

Data： 用户自定义数据，会写入event中。

* 1. take\_order(maker: address, orderidx: u64, idxa: u64, amounta: u64, idxb: u64, amountb: u64, data: vector<u8>)

币币交易的taker的取单操作，**注意：idxa，amounta，idxb，amountb必须和挂单操作一样。**

Orderidx：挂单编号

Idxa：maker卖出的币种编号

Amounta：maker卖出的币种数量

Idxb：maker买入的币种编号

Amountb：maker买入的币种数量

Data： 用户自定义数据，会写入event中。

* 1. move\_owner(tokenidx: u64, new\_owner: address, data: vector<u8>)

转移铸币权，owner才能调用。

Tokenidx： 币种编号

New\_owner: 新的铸币人

Data： 用户自定义数据，会写入event中。

* 1. balance(tokenidx: u64) ：u64

获取用户余额。

Tokenidx： 币种编号

* 1. token\_count() ：u64

获取当前已经创建的币种数量。

下面是各函数产生的event对应的etype的值列表：

0： publish

1： create\_token

2： mint

3： transfer

4： make\_order

5： cancel\_order

6： take\_order

7： move\_owner

1. **使用流程实例**
   1. 人物介绍

S: 超级管理员

Ta：币种a，tokenidx为0

Tb：币种b，tokenidx为1

Oa：币种a的owner

Ob：币种b的owner

U1：普通用户1

U2: 普通用户2

* 1. 使用流程：
     1. 所有人物角色分别获取平台币
     2. S将合约publish module发布到链上
     3. S调用publish进行合约注册
     4. S调用create\_token发行币种Ta，Tb，并分别指定owner Oa，Ob
     5. Oa，Ob，U1，U2 分别调用publish进行合约注册
     6. Oa调用mint给U1铸Ta币种的100块钱
     7. Ob调用mint给U2铸Tb币种的100块钱
     8. U1调用transfer给U2转账Ta币种的50块钱
     9. U2调用transfer给U1转账Tb币种的50块钱
     10. 现在U1有Ta 50块，Tb 50块，U2也是一样
     11. U1调用make\_order币币交易挂单卖出Ta 10块，买入Tb 20块, 调用参数为make\_order(Ta, 10, Tb, 20, data)
     12. U2调用take\_order去取U1挂的单， 成功后：U1有Ta 40块，Tb 70块， U2有Ta 60块，Tb 30块。