# Violas move合约说明文档

1. 铸币（mint） // mint.mvir
   1. 参数说明
      1. payee: address // 接收人地址
      2. amount: u64 // 铸币的数量
2. 用户新资产token的注册 // publish.mvir
   1. 参数说明：无
3. Token的转让 // transfer.mvir
   1. 参数说明
      1. payee: address // 接收人地址
      2. amount: u64 // 转让的数量
4. 交易所token挂单 // make.mvir
   1. 参数说明
      1. amount: u64 // 卖出token的数量
      2. price: u64 // 买入token的数量
      3. flag: u64 // 卖出挂单token的类型: 0:稳定币，1:平台币
5. 交易所token撤单 // cancel.mvir
   1. 参数说明：
      1. flag: u64 // 卖出挂单token的类型: 0:稳定币，1:平台币
6. 交易所取单 // take.mvir
   1. 参数说明
      1. maker: address // 卖出token的挂单人的地址
      2. flag: u64 // 卖出挂单token的类型: 0:稳定币，1:平台币

**注意事项：**

* Github地址： <https://github.com/palliums-developers/exchange-matchengine/tree/master/violas_move>
* 当前公司部署的violas链上我已经部署了3种新发行的token，地址如下：

7257c2417e4d1038e1817c8f283ace2e1041b3396cdbb099eb357bbee024d614

c20bca7777142dc8ada8ea6ced1c02519aac0b1f0d27149eb0bbe02ee22f5cd9

6fa7359891bef85cc7aa12787b17e85d3f75c78525ffd39cb77eb3453eb3bb75

* Github上mv类型的文件是libra的compiler编译后的字节码文件

{"code":[76,73,66,82,65,86,77,10,1,0,7,1,74,0,0,0,4,0,0,0,3,78,0,0,0,6,0,0,0,13,84,0,0,0,4,0,0,0,14,88,0,0,0,2,0,0,0,5,90,0,0,0,27,0,0,0,4,117,0,0,0,64,0,0,0,8,181,0,0,0,11,0,0,0,0,0,1,1,0,2,0,1,3,0,2,0,0,0,3,0,6,60,83,69,76,70,62,6,68,84,111,107,101,110,4,109,97,105,110,7,112,117,98,108,105,115,104,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,114,87,194,65,126,77,16,56,225,129,124,143,40,58,206,46,16,65,179,57,108,219,176,153,235,53,123,190,224,36,214,20,0,1,0,1,0,2,0,19,1,0,2],"args":[]}

**Code字段对应的就是客户端程序需要的字节码， 所有的mv文件都是用第一个token的地址来生成的（7257c2417e4d1038e1817c8f283ace2e1041b3396cdbb099eb357bbee024d614），当客户端想要组其他token的交易时， 请将mv文件中的7257c24....的这个地址替换成对应的token地址即可。**

* 合约挂单操作只能挂新发行的token
* **所有的新发行资产的token的操作第一步必须执行publish操作，也就是进行新发行资产的注册， 比如，你给某人转让某种新发行的token， 他如果没有做过这个token的publish，转让是会失败的。**
* 交易产生的Event说明：

*etype: u64, // event类型，具体类型后面说明*

*sender: address, // 交易发送人地址*

*receiver: address, // 交易接收人地址*

*token: address, // 交易的token的地址*

*amount: u64, // 交易的数量*

*price: u64, // 交易所交易时平台币的数量*

Event类型：

*0： publish 新资产token的注册*

*1: mint 新资产token的铸币*

*2: transfer 新资产token的转让*

*3： make 新资产token的挂单*

*4： cancel 新资产token的撤单*

*5： take 新资产token的取单*

*6： make2 平台币token的挂单*

*7： cancel2 平台币token的撤单*

*8： take2 平台币token的取单*

**0,3,4这三种操作时只会产生一条event， 1,2,5这三种操作会针对发送人和接收人各产生一条event共两条event。**

* **州长操作流程：**
  + 获取平台币以用做gas费用
  + 用州长自己的账户A将自己发行的稳定币合约A发布上链(libra client程序： dev publish ...)
  + 州长用账户A调用合约A的publish.mvir函数进行token注册
  + 用户B获取平台币用作gas费用
  + 用户B调用合约A的publish.mvir进行token注册
  + 州长用账户A调用合约A的mint.mvir脚本给来铸币并把币存在用户B账户下。
  + 用户C获取平台币用作gas费用
  + 用户C调用合约A的publish.mvir进行token注册
  + 用户B调用合约A的transfer.mvir脚本来给用户C进行转账
* **普通用户操作流程**
  + 用户B获取平台币用作gas费用
  + 用户B调用合约A的publish.mvir进行token注册
  + 用户B接收别人给他的稳定币token转账或者转账给其他人(transfer脚本)