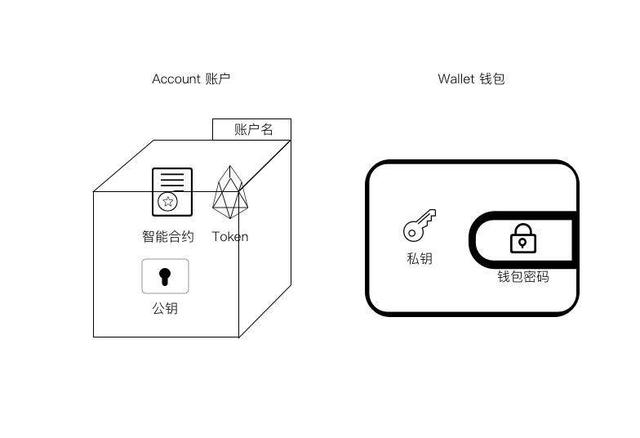
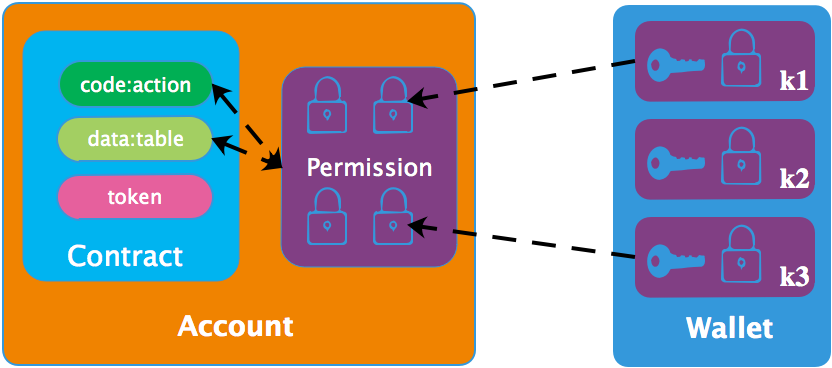
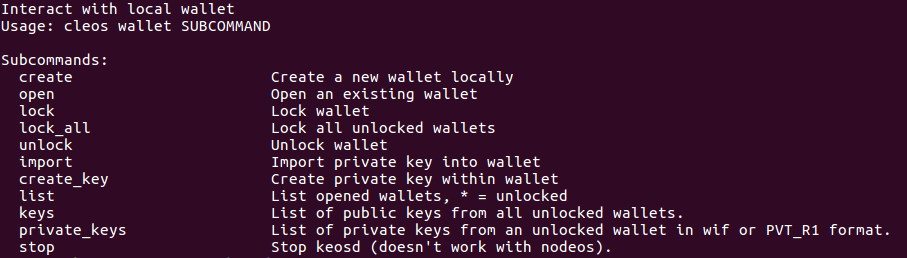
在开始讲解钱包之前，我们先看下面的两幅有关钱包与账户的之间关系的图片，从两幅图片中，可以得知，所谓钱包的作用就是存放密钥的，其密钥是为该账户下的各种需要加密服务的操作服务的，如 交易，签名，出块等等。





wallet操作的所有子命令，主要有创建，打开，列出，加锁，解锁，导入key， 显示public keys，显示private keys，如下图所示：



**创建**：*$ cleos wallet create -n rabbit*

*Creating wallet: rabbit*

*Save password to use in the future to unlock this wallet.*

*Without password imported keys will not be retrievable.*

*"PW5KAUEyoTRXZkGFEbWhFEa7p2K4JJDej45MZBAEewxzubtHEZQmh"*

*上面创建了一个名字为rabbit的钱包，在返回中，可以看到有一串密码，该密码需要我们保存下来，切勿丢失，以后对钱包解锁，导入keys等操作都需要密码才能进行。*

**列出**：*$ cleos wallet list*

*$ cleos wallet list*

*Wallets:*

*[*

*"rabbit \*"*

*]*

*从返回结果可以看到，名为rabbit的钱包，后面的’\*’表示钱包是未锁定状态。*

**加锁**： *$ cleos wallet lock -n rabbit*

*Locked: rabbit*

*$ cleos wallet list*

*Wallets:*

*[*

*"rabbit"*

*]*

*从上面的操作可以看到’\*’已经消失，钱包处于锁定状态。*

**解锁**： *$ cleos wallet unlock -n rabbit*

*password:*

*Unlocked: rabbit*

*在解锁的过程中需要输入创建钱包时所返回的密码。*

我们在进行key操作的时候首先要创建key，如果不创建的话，钱包里会有一对默认的key。

**创建key**：*$ cleos create key*

*Private key: 5JMavdjxfTDgzqjCGfNos4KAbFAyT6DCmwx7neJBFXbNWiXHDMR*

*Public key: EOS7MJc2bfRZxNYkYt27EUBz5KugBswXTFcFEKfTKzXmjVfVPSXrx*

*公钥都是以EOS开头的。*

**导入key**： *$ cleos wallet import 5JMavdjxfTDgzqjCGfNos4KAbFAyT6DCmwx7neJBFXbNWiXHDMR -n rabbit*

*imported private key for: EOS7MJc2bfRZxNYkYt27EUBz5KugBswXTFcFEKfTKzXmjVfVPSXrx*

**列出public keys**： *$ cleos wallet keys*

*[*

*"EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV",*

*"EOS7MJc2bfRZxNYkYt27EUBz5KugBswXTFcFEKfTKzXmjVfVPSXrx"*

*]*

**列出private keys**：*$ cleos wallet private\_keys -n rabbit*

*password:*

*[[*

*"EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV",*

*"5KQwrPbwdL6PhXujxW37FSSQZ1JiwsST4cqQzDeyXtP79zkvFD3"*

*],[*

*"EOS7MJc2bfRZxNYkYt27EUBz5KugBswXTFcFEKfTKzXmjVfVPSXrx",*

*"5JMavdjxfTDgzqjCGfNos4KAbFAyT6DCmwx7neJBFXbNWiXHDMR"*

*]*

*]*

*在列出private keys时，需要输入创建账户时返回的密码。*

以上就是对钱包的基本操作，至此，我们已经完成了钱包的创建以及将密钥放入钱包以保存。

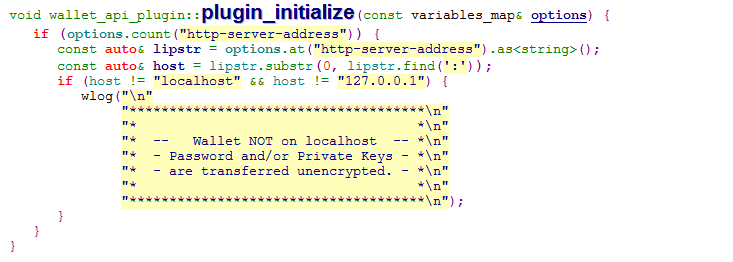
**Wallet plugin分析**

Wallet有关插件，主要有wallet\_api\_plugin, wallet\_plugin以及提供网络功能的http\_plugin。

Wallet\_api\_plugin包含Wallet\_api\_plugin.h, 主要向客户端提供wallet api，其实现方式是通过将api指针与curl进行map映射。在客户端发送相应的curl时，以调用相应的api，并返回结果。

Wallet\_plugin,主要包括Wallet\_plugin.h，Wallet\_manager.h，Wallet.h，其中Wallet\_plugin.h主要对curl中的option选项进行相应的操作； Wallet.h 中是完成钱包各项操作的具体实现；Wallet\_manager.h像一个中间层，起连接作用，向上面向Wallet\_api\_plugin.h，向下面向Wallet.h。

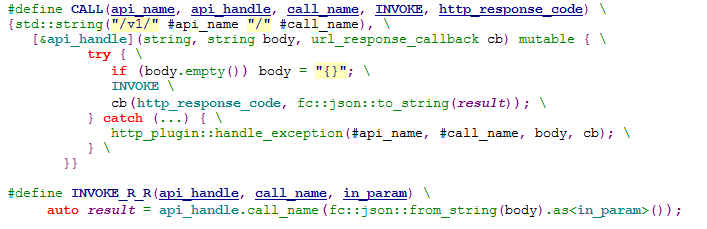
1. Wallet\_api\_plugin初始化：

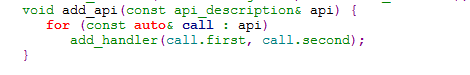


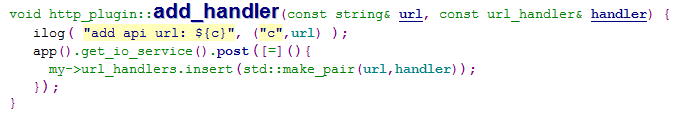
1. Wallet\_api\_plugin启动：



其中get\_wallet\_manager()获取wallet\_manager句柄，add\_api是http\_plugin的接口，其作用是将curl与具体功能实现接口添加到map映射表，当http\_plugin模块收到相应curl后，调用相应的功能，并返回结果。下面是具体代码：

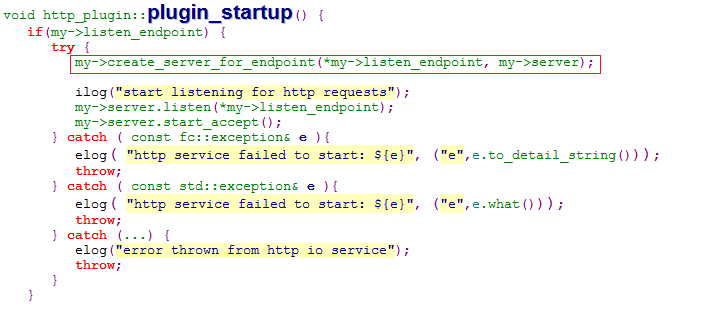


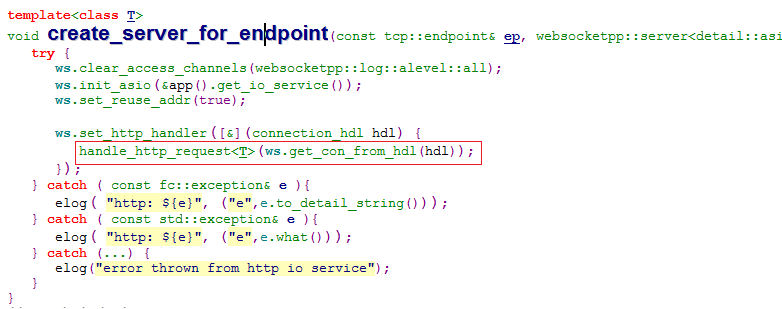




至此，curl与handler添加进入map(url\_handles)。

下面我们看一下以wallet create为例，跟踪一下服务器端的代码：在开始之前，先看一下http\_plugin的启动。







从上面红框内可以看到当http\_plugin启动后，将监听端口，当有http request到达时，获取uri, 查询url\_handle map, 以调用相应的功能，最后通过send\_http\_response将结果发送到客户端。

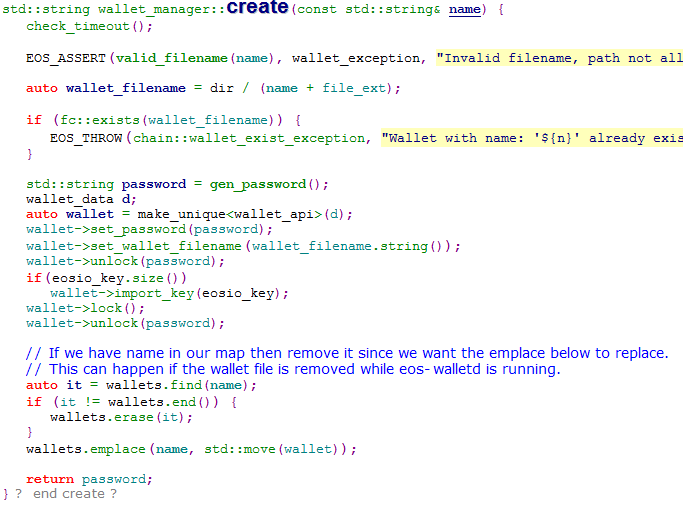
通过下面最终create函数，将钱包创建成功，其文件放在~/eosio-wallet/目录下。

*$ cat rabbit.wallet*

*{*

*"cipher\_keys": "459e729d74756b0dfc80534e19fee1211a7be31ee44d0b3a5dd4d8072a56bf046c902d52f55a286c471c6704d3ecd6a0374224db22ed726308ff2fd412ad5a165092311e2642d3de4294e8ded6656dc1bb97a4a0c34da9545d033e63299557c40fe04f9cf02a24086e3dda70ec9303423ba3c2fef92df4baf72d8543e61cccce7761d27ecc99d92d84b7bfa7c952be53754f1a93506f81e38425ae31e0046d5704bbee3038f3e751e1219d8cc7c7ee93b47f88bcc525d0d3842a1fdf12036d5b60374a36f21667cbf01ced0f7f9b5d3b"*

*}*



Request：

$ curl http://localhost:8888/v1/wallet/create -X POST -d '"rabbit"'

Response：

PW5KFWYKqvt63d4iNvedfDEPVZL227D3RQ1zpVFzuUwhMAJmRAYyX

对于钱包操作的curl，以及返回格式可以参考：

[*https://blog.csdn.net/weixin\_38234890/article/details/80339567*](https://blog.csdn.net/weixin_38234890/article/details/80339567)