

SEED程序使用手册

A. 初始化和进入程序

1. 在terminal中进入到SEED程序所在目录下。
2. 在该目录下输入命令：`./seed --path=./workpath/ --port=8899`，path参数用于指定seed工作目录（`--path=./workpath`后面必须加/），未指定则默认为当前目录。port参数用于指定端口号，未指定则默认为8899（即可以直接输入`./seed`）。
3. 首次启动SEED程序会被要求初始化，如下图所示：

```
➔ seed ./seed --path=./ --port=8899
data path: "./", port: "8899".
lock file successfully
The seed has not been initialized, please type "yes" to continue.
-> 
```

4. 输入yes后，会被要求设置密码，如下图所示：

```
The seed has not been initialized, please type "yes" to continue.
-> yes
Input password first time.
-> 
```

5. 输入要为SEED程序设置的密码（最好足够复杂）。点击enter键，需要再输入一遍密码以确认，之后点击enter键，初始化完成，如下图所示。

```
Input password again.
->
Initialization finished. Please start the server again.
```

6. 输入命令：`./seed --path=./workpath/ --port=8899` 重新启动程序。会被要求输入密码，必须和初始化时设置的密码相同。

```
➔ seed ./seed --path=./ --port=8899
data path: "./", port: "8899".
Please input passowrd.
-> 
```

7. 输入密码正确后，SEED程序启动，启动成功后如下图所示。

```
→ seed ./seed --path=./ --port=8899
data path: "./", port: "8899".
Please input passowrd.
->
Password is correct.
config.json md5[99914b932bd37a50b983c5e7c90ae93b] pass check
data.json md5[99914b932bd37a50b983c5e7c90ae93b] pass check
seed.json md5[60bec01aec03677dd8bfc74052294461] pass check
lock file successfully
Successfully recover the seed, now starting the seed server :)
Authorized client is: [127.0.0.1].

  ____  _
 / ___\/___\ /___\ /___\
/___\ /___\ /___\ /___\
/___\ /___\ /___\ /___\ v3.2.6
High performance, minimalist Go web framework
https://echo.labstack.com
-----0/-----
                                0\
⇒ http server started on [::]:8899
```

B. 备份程序数据

当SEED程序初始化完成后，指定工作目录下会生成一个seed.json文件，该文件中包含SEED程序初始化设置密码生成的哈希值，此哈希值用来验证密码是否正确以确保SEED程序被密码保护。此外seed.json还包含一个seed密文，此seed密文用来和密码一起解密出seed明文，seed明文用来验证备份是否成功。seed.json文件示意图如下。

```
{
  "hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
  "seed": "laJpORTjLNismX6cCnzDlwGrOjIG+BCIR2+1rVnyxpr3EXIMuL4lQhtUFmsxVl2FILSO1AK+GSIsZ1p8kFKsWQE0+hGgA6s6DGvdWjhRA="
}
```

另外还会生成seed.json.md5的隐藏文件，当有人手工修改任何json或者md5文件中的内容，都会导致服务无法启动。

具体备份方法如下：

1. 按照A所示步骤完成seed程序初始化。
2. 在SEED程序所在目录下输入命令：./seed --path=./workpath/ --verify，会被要求输入一次密码，必须和初始化密码完全相同。
3. 输入密码正确后，将示意图中所示的seed（seed明文）和seed MD5备份。

```
➔ seed ./seed --path=./ --verify
data path: "./", port: "8899".
Please input passowrd.
->
Password is correct.
config.json md5[99914b932bd37a50b983c5e7c90ae93b] pass check
data.json md5[99914b932bd37a50b983c5e7c90ae93b] pass check
seed.json md5[60bec01aec03677dd8bf74052294461] pass check
You are verifying the seed now.
Seed:      1a97bc44.....2d6fd8d3
Seed MD5:  a1b724da6ede58cf158eaa8163112957
Verification finished. Please start the server again.
```

4. 将SEED程序工作目录下的seed.json文件和seed.json.md5备份。
5. 将备份的seed.json和seed.json.md5文件拷贝到备份电脑中SEED程序（未被初始化的，如果已被初始化应先删除其工作目录下的seed.json文件）的工作目录下。
6. 在备份电脑中的terminal里进入到SEED程序所在目录下，输入命令：./seed --path=./workpath/ --verify，会被要求输入密码，必须和被备份的SEED程序设置的初始化密码完全相同。
7. 输入密码正确后，将verify出现的seed和seed MD5，与被备份的seed和seed MD5做对比。
8. 若结果完全相同，说明备份成功。

C. 设置自定义数据

1. 按照A所示步骤完成SEED程序初始化。
2. 在SEED程序所在目录下，输入命令：`./seed --path=./workpath/ --data`，会被要求输入一次密码，必须和被备份的SEED程序设置的初始化密码完全相同。
3. 输入密码正确后，会被要求输入数据名，在此处输入key（配置键值），配置具体格式和说明请参照SEED配置格式.pdf。
4. 输入数据名后，会被要求输入数据，具体格式和说明请参照SEED配置格式.pdf。
5. 如果设置CYB数据则根据不同钱包类型有以下三种模式：
 - a. 云钱包模式：将热钱包账户，热钱包账户密码和冷钱包账户一起设置在数据里，最后加上模式类型password或者不加（password拼写必须相同并小写），用/分隔开，格式为：热钱包账户/热钱包账户密码/冷钱包账户/password，或热钱包账户/热钱包账户密码/冷钱包账户（例如：apdae0/apdae1APDAE2eadpa3/apdae1/password或apdae0/apdae1APDAE2eadpa3/apdae1）。
 - b. 钱包模式+私钥：将热钱包账户，热钱包账户私钥（账户配置-高级设定-权限-私钥查看器）和冷钱包账户一起设置在数据里，最后加上模式类型prikey（prikey拼写必须相同并小写），用/分隔开，格式为：热钱包账户/热钱包账户私钥/冷钱包账户/prikey（例如：apdae0/5JM5t8kkisBfiLKtRaAh2cioJ4AAiFSbp77WGAdStEGmSSkkHcT/apdae1/prikey）。
 - c. 钱包模式+备份bin：将热钱包账户，热钱包账户密码和冷钱包账户一起设置在数据里，最后加上模式类型bin（bin拼写必须相同并小写），用/分隔开，格式为：热钱包账户/热钱包账户密码/冷钱包账户/bin（例如：apdae0/apdae1APDAE2eadpa3/apdae1/bin）。另外，热钱包账户的备份bin文件需被放置在瑶池主程序目录下，并在瑶池的config配置中配置该文件路径（默认路径：secret/cybex-hot.bin）。
6. 瑶池启动区块链的数据必须全部设置，缺一不可，被设置的数据统一以密文形式被存储在seed工作目录下的data.json文件中，同时也会生成data.json.md5的隐藏文件，当有人手工修改任何json或者md5文件中的内容，都会导致服务无法启动。
7. 数据设置完毕，输入命令：`./seed --path=./workpath/ --port=8899`，重新启动SEED程序。

```
➔ seed ./seed --path=./ --data
data path: "./", port: "8899".
Please input passowrd.
->
Password is correct.
config.json md5[99914b932bd37a50b983c5e7c90ae93b] pass check
data.json md5[99914b932bd37a50b983c5e7c90ae93b] pass check
seed.json md5[60bec01aec03677dd8bfc74052294461] pass check
You are set the data now.
Please input the name first.
-> BTC
Please input the data.
-> n15W1MhkVsQvPySETqPHKgTvKDJv8UjaQn
BTC:n15W1MhkVsQvPySETqPHKgTvKDJv8UjaQn
Setting data finished. Please start the server again.
```

BTC设置数据示意图

```
"CYB": {
  "HotWallet": {
    "Address": "",
    "Secret": "",
    "Mode": "",
    "BinPath": "secret/cybex-hot.bin"
  },
  "ColdWallet": {
    "Address": ""
  },
  "RescanMode": "block",
  "HighWaterLevel": 3000,
  "SweepTo": 2500,
  "LowWaterLevel": 1000
}
```

CYB备份bin文件在瑶池目录下的路径设置

D. 查询已设置数据

1. 按照C所示步骤设置完成每个币种相关数据。
2. 在SEED程序所在目录下，输入命令：`./seed --path=./workpath/ --list`，会被要求输入一次密码，必须和被备份的SEED程序设置的初始化密码完全相同。
3. 输入密码正确后，会看到已设置的所有数据明细，如下图所示。

```
➔ seed ./seed --path=./ --list
data path: "./", port: "8899".
Please input passowrd.
->
Password is correct.
config.json md5[99914b932bd37a50b983c5e7c90ae93b] pass check
data.json md5[d80c6a16c73861e5b10b88f60b629339] pass check
seed.json md5[60bec01aec03677dd8bfc74052294461] pass check
The data list is:
BTC: "n15W1MhkVsQvPySETqPHKgTvKDjv8UjaQn"
CYB: "apdae0/apdae1APDAE2eadpa3/apdae1"
ETH: "0x6b0135684722025ee4b52fae9e1e6cdbcea4aa2d"
The config list is:
```

E. 设置访问权限

为保证SEED程序安全并有一定的拓展性，程序必须设置有权限访问的IP地址，但同时SEED程序的管理员有权根据实际情况修改被允许访问SEED程序的IP地址。SEED程序初始化后默认只有本机可以访问，如果管理员选择修改则原IP地址被新IP地址覆盖。SEED程序被初始化后会在同目录下生成一个config.json文件，用下述方法设置的参数都会被存储在该文件中。具体修改访问权限方法如下：

1. 按照A所示步骤完成SEED程序初始化。
2. 在SEED程序所在目录下，输入命令：`./seed --path=./workpath/ --config`，会被要求输入一次密码，必须和被备份的SEED程序设置的初始化密码完全相同。
3. 输入密码正确后，会被要求输入参数名，若此处是修改被允许访问SEED程序的IP地址则该处必须填client（小写字母）。
4. 输入参数名后，会被要求输入新的被允许访问SEED程序的IP地址，多个IP地址用逗号分隔。被设置的数据统一以密文形式被存储在seed工作目录下的config.json文件中，同时也会生成config.json.md5的隐藏文件，当有人手工修改任何json或者md5文件中的内容，都会导致服务无法启动。

```
→ seed ./seed --path=./ --config
data path: "./", port: "8899".
Please input passowrd.
->
Password is correct.
config.json md5[99914b932bd37a50b983c5e7c90ae93b] pass check
data.json md5[d80c6a16c73861e5b10b88f60b629339] pass check
seed.json md5[60bec01aec03677dd8bfc74052294461] pass check
You are set the config now.
Please input the name first.
-> client
Please input the config.
-> 127.0.0.1,11.22.33.44,1.2.3.4
client:127.0.0.1,11.22.33.44,1.2.3.4
Setting config finished. Please start the server again.
```

5. 设置成功，输入命令：`./seed --path=./workpath/ --port=8899`，重启SEED程序。
6. 重启程序后，会看到“Authorized client is:（新IP地址）”，此处可用来验证修改是否成功。


```

→ seed ./seed --path=./ --port=8899
data path: "./", port: "8899".
Please input passowrd.
->
Password is correct.
config.json md5[a4dd4ca3c97edf5182c1f7ac0a322549] pass check
data.json md5[d80c6a16c73861e5b10b88f60b629339] pass check
seed.json md5[60bec01aec03677dd8bfc74052294461] pass check
lock file successfully
Successfully recover the seed, now starting the seed server :)
Authorized client is: [127.0.0.1 11.22.33.44 1.2.3.4].

```

```

      _ _ _ _ _
     /  _  / _ _  /  _ _ _
    /  _//  _/  _ \  _ _ \
   / _ _/\_ _/ _// _/\_ _/ v3.2.6
High performance, minimalist Go web framework
https://echo.labstack.com

```

7. 此外还有另一方法可以验证修改，在SEED程序所在目录下，输入命令：./seed --path=./workpath/ --list，会看到“The config list is: client: （新IP地址）”。

```

→ seed ./seed --path=./ --list
data path: "./", port: "8899".
Please input passowrd.
->
Password is correct.
config.json md5[a4dd4ca3c97edf5182c1f7ac0a322549] pass check
data.json md5[d80c6a16c73861e5b10b88f60b629339] pass check
seed.json md5[60bec01aec03677dd8bfc74052294461] pass check
The data list is:
BTC: "n15W1MhkVsQvPySETqPHKgTvKDJv8UjaQn"
CYB: "apdae0/apdae1APDAE2eadpa3/apdae1"
ETH: "0x6b0135684722025ee4b52fae9e1e6cdbcea4aa2d"
The config list is:
client: "127.0.0.1,11.22.33.44,1.2.3.4"

```

F. 修改进入程序密码

方法如下：

1. 在SEED程序所在目录下，输入命令：`./seed --path=./workpath/ --password`，会被要求输入一次密码，必须和被备份的SEED程序设置的初始化密码完全相同。
2. 接着要求输入两次新密码。
3. 程序会将旧seed.json备份到seed.json.last，.seed.json.md5备份到seed.json.md5.last，并生成新的seed.json和.seed.json.md5，data和conifg同理，此时密码修改完毕。

```
→ seed ./seed --path=./ --password
data path: "./", port: "8899".
Please input passowrd.
->
Password is correct.
config.json md5[a4dd4ca3c97edf5182c1f7ac0a322549] pass check
data.json md5[523e2facbc75dbe466bfc2802053340] pass check
seed.json md5[60bec01aec03677dd8bfc74052294461] pass check
lock file successfully
You are change the password now.
Input new password first time.
->
Input new password again.
->
seed.json move to seed.json.last
data.json move to data.json.last
config.json move to config.json.last
.seed.json.md5 move to seed.json.md5.last
.data.json.md5 move to data.json.md5.last
.config.json.md5 move to config.json.md5.last
Change password succeeded, Please start the server again.
```

G. seed程序批量导入data

可以对SEED程序的data进行批量导入，文件test.txt中包含冒号分割的数据，每行是一组data数据。如：

```
1 abc:xxx
2 cyb:sasd
3 asdasd:asdasd asd asda
4 aas:dad
```

具体操作如下：

1. 在SEED程序所在目录下，输入命令： `./seed --path=./workpath/ --import test.txt`，这里的test.txt是任意导入文件名。会被要求输入一次密码，必须和被备份的SEED程序设置的初始化密码完全相同。
2. 如果文件解析正确，会打出一共要导入的数据条数和覆盖的数据条数，打入 yes 确认导入。导入结束

```
➔ seed ./seed --path=./ --import data.txt
data path: "./", port: "8899".
Please input passowrd.
->
Password is correct.
config.json md5[a4dd4ca3c97edf5182c1f7ac0a322549] pass check
data.json md5[d80c6a16c73861e5b10b88f60b629339] pass check
seed.json md5[b8039705551e50084168b9707375b7f1] pass check
You will import 4 datas,overwrite 0 datas,confirm by type 'yes'
-> yes
import data finished. Please start the server again.
```