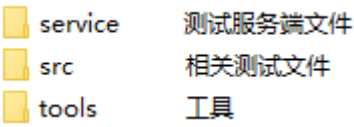


Neo python 测试框架基础使用介绍

一、目录结构

1. 测试服务端及相关工具文件如图 1:



A directory tree diagram showing three folders: 'service', 'src', and 'tools'. Each folder is represented by a yellow folder icon. To the right of each folder icon is its name and a descriptive label in Chinese.





 service	测试服务端文件
 src	相关测试文件
 tools	工具

图 1

2. 客户端测试框架如图 2:



A directory tree diagram showing a mix of folders and files. Folders are represented by yellow folder icons, and files by white document icons. Each item is followed by its name and a descriptive label in Chinese.



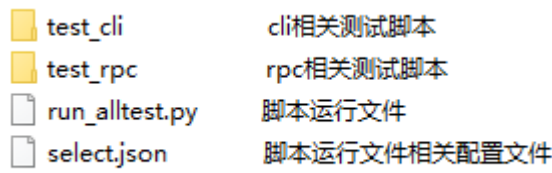
 api	测试脚本相关API
 monitor	检测恢复测试环境
 neo	钱包控制文件
 resource	自检文件相关
 test	测试脚本
 utils	测试框架代码
 config.json	配置文件

图 2

3. 客户端测试用例及相关工具如图 3:



A directory tree diagram showing folders and files. Folders are represented by yellow folder icons, and files by white document icons. Each item is followed by its name and a descriptive label in Chinese.





 test_cli	cli相关测试脚本
 test_rpc	rpc相关测试脚本
 run_alltest.py	脚本运行文件
 select.json	脚本运行文件相关配置文件

图 3

二、测试环境搭建及运行测试脚本

使用自检文件配置：

1. 在 GitHub 上下载 neo 测试相关文件，下载文件 neo-test-master 打开后如图 1 所示。其中 service 中有测试服务工具。在 src 中包含测试脚本、测试框架相关代码工具、自检文件、检测恢复等文件，如图 2 所示。

2. 首先将下载的 neo-test-master 配置到共识服务器及外部节点的服务器/root/neo/neo-test-master/下，然后按照以下步骤安装所需 python 模块：

```
$ cd /root/neo/neo-test-master/
```

```
$ chmod 777 init.sh
```

```
$ ./ init.sh
```

3. 接着共识服务器启动 neo-test-master 下的 service(默认是启动 4 个服务，分别对应 4 个 NEO 共识节点)，按以下步骤启动：

```
$ cd /root/neo/neo-test-master/service
```

```
$ chmod 777 run.sh
```

```
$ ./ run.sh
```

4. 再在外部节点服务器启动 neo-test-master 下的 service(默认是启动 2 个服务，分别对应 2 个 NEO 共识节点)，按以下步骤启动：

```
$ cd /root/neo/neo-test-master/service
```

```
$ chmod 777 run_output.sh
```

```
$ ./ run_output.sh
```

5. 使用自检文件核对并修改测试环境。自检文件会根据用户配置的 Config.json 文件，将上传 NEO 节点，初始化区块链，重启共识节点一系列自动化操作。

在 neo-test-master/tools/路径下包含了自检文件 (init_selfchecker.py)，在 neo-test-

master\src\resource 下有自检文件中需要使用的最新文件。在实现自检之前需要修改配置文件中的内容，文件位置：/root/neo/neo-test-master/src/config.json 配置修改示例如下：

```
{
  "NODES": [
    {
      "ip": "47.244.25.229",
      "consensusnode": true,
      "sshuser": "root",
      "sshpwd": "NEO@neo.org",
      "rpcport": 10003,
      "listenport": 10001,
      "stport": 23606,
      "walletname": "1.json",
      "path": "/root/neo/nodes/node1/neo-cli.dll"
    },
    {
      "ip": "47.244.25.229",
      "consensusnode": true,
      "sshuser": "root",
      "sshpwd": "NEO@neo.org",
      "rpcport": 20003,
```

```

        "listenport" : 20001,
        "stport" : 23616,
        "walletname" : "2.json",
        "path" : "/root/neo/nodes/node2/neo-cli.dll"
    },
    {
        "ip": "47.244.25.229",
        "consensusnode":true,
        "sshuser": "root",
        "sshpwd": "NEO@neo.org",
        "rpcport" : 30003,
        "listenport" : 30001,
        "stport" : 23626,
        "walletname" : "3.json",
        "path" : "/root/neo/nodes/node3/neo-cli.dll"
    },
    {
        "ip": "47.244.25.229",
        "consensusnode":true,
        "sshuser": "root",
        "sshpwd": "NEO@neo.org",
        "rpcport" : 40003,
        "listenport" : 40001,
        "stport" : 23636,
        "walletname" : "4.json",
        "path" : "/root/neo/nodes/node4/neo-cli.dll"
    },
    {
        "ip": "47.244.25.229",
        "consensusnode":false,
        "sshuser": "root",
        "sshpwd": "NEO@neo.org",
        "rpcport" : 50003,
        "listenport" : 50001,
        "stport" : 23646,
        "walletname" : "5.json",
        "path" : "/root/neo/nodes/node5/neo-cli.dll"
    }
],
"WALLETS": [
    "1.json",
    "2.json",
    "3.json",
    "4.json",

```

```

        "5.json"
    ]
}

```

以上 Config 是提供给测试脚本，测试框架，自检脚本使用，具体 Config 解释：

```

{
    "NODES": [
        {
            "ip": "测试节点的 IP 地址",
            "consensusnode": "是否共识节点",
            "sshuser": "测试节点服务器 ssh 登录名",
            "sshpwd": "测试节点服务器 ssh 密码",
            "rpcport": "测试节点 RPC 端口",
            "listenport": "监听端口",
            "stport": "测试节点 Service Tool 端口",
            "walletname": "测试节点使用的 wallet 的名称",
            "path": "测试节点 neo-cli.dll 的绝对路径"
        }
    ],

    "WALLETS": [
        测试节点使用的 wallet 名称，统一存放在 neo-test-master/src/resource/wallet 下
    ]
}

```

同时，/root/neo/neo-test-master/src/rscource 下文件也需要换成对应的 protocol.json/config.json。

neo-test-master/tools/路径下包含了自检文件（init_selfchecker.py）

自检步骤解释如下：

1. 为了防止出现数据异常、丢失等问题，需要关闭所有节点（stop_all nodes）。
2. 关闭所有节点之后，需要清除掉节点（clear_nodes）。在此之前，需要进行备份的请及时进行备份，以防数据丢失。清除节点路径/root/neo/nodes。
3. 拷贝节点（copy_node），将neo-test-master \src \resource 下的 neo-cli.tar.gz 拷贝至路径 /root/neo/nodes/然后解压至各个 node 文件夹中,将 neo-cli2.tar.gz 拷贝至最后一个节点中（如有）。之后把 acc 文件拷贝至路径/root/neo/nodes/中。路径\neo-test-master \src \resource\nodes\node1、...下的 config.json、protocol.json,以及路径\neo-test-master \src \resource\wallet 下的 json 文件拷贝至/root/neo/nodes 这里的 config.json、protocol.json、wallet.json 已经在服务器上配置好。
4. 启动节点(start_nodes)。
5. 检查核对节点(check_nodes)。

最后根据如下操作在外部节点服务器上完成自检：

```

$ cd /root/neo/neo-test-master/tools/
$ python3 init_selfchecker.py

```

脚本测试步骤如下：

测试脚本位于/root/neo/neo-test-master/src/test 下，路径下文件内容如下：





使用 run_alltest.py 可以实现选择性测试相关用例，select.json 为选择性执行测试脚本的配置文件。

在进行测试之前可以先获取所有脚本的执行权限，步骤如下：

```
$ cd /root/neo/neo-test-master/src/
```

```
$ chmod -R 777 test
```

```
$ cd /root/neo/neo-test-master/src/test $ python3 run_alltest.py #执行所有 case
```

	test_cli	cli相关测试脚本
	test_rpc	rpc相关测试脚本
	run_alltest.py	脚本运行文件
	select.json	脚本运行文件相关配置文件

另外，如若无需测试所有的 case，可以按照如下规则调用 run_alltest.py 有选择地执行 case：

在 run_alltest 中导入了 src/monitor 下 monitor.py 中的 TestMonitor，monitor 可以在执行 case 时监测环境中的未知错误和恢复测试环境。调用 run_alltest 执行 case 时，可以根据参数选择执行需要测试的 case，当然首先需要到 run_alltest 所在路径：

```
$ cd /root/neo/neo-test-master/src/test
```

调用 run_alltest 时，其后可以跟-c(--config)、-t(--type)、-f(--filter)、-m(--monitor)、-e(--exclude)。

1) -c 后跟配置文件，一般 neo-test 为 select.json。将 select.json 中修改如下：

```
{
    "test_rpc": true,
    "test_cli": false
}
```

在进行如下调用时便可选择只执行 RPC 相关测试：

```
$ cd /root/neo/neo-test-master/src/test
```

```
$ python3 run_alltest.py -c select.json
```

2) -t 后跟 case 类型："base"，"normal"，"abnormal"。-t 为"base"时便可选择所有 case 中方法名中包含_base 的 case，"abnormal"时便可选择所有 case 中方法名中包含_abnormal 的 case 即期望结果为异常的 case，但注意"normal"时是选择所有 case 中方法名中包含_normal 和_base 的所有 case 即期望结果为正常的 case，示例如下：

```
$ cd /root/neo/neo-test-master/src/test
```

```
$ python3 run_alltest.py -t "base"
```

3) -f 后跟具体的 case

4) -m 后跟"0"或"1"，0 代表不启用 monitor，1 代表启用 monitor，调用 run_alltest 时无-m 时默认启用 monitor，其调用规则如下：

```
$ cd /root/neo/neo-test-master/src/test
```

```
$ python3 run_alltest.py -m "1" 或 python3 run_alltest.py -m 1
```

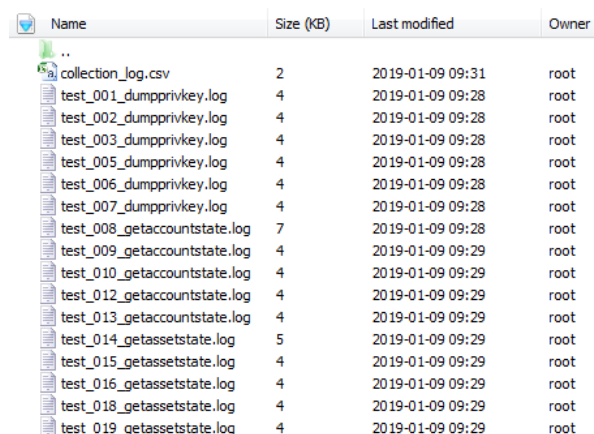
5) -e 后跟"类名.方法名"与-c 配合，可以用于执行-c 中选定的测试例集合中排除-e 中的那些测试例。

6) 在/neo-test-master/src/logs 文件夹下，存放的是运行测试用例后记录的 log，文件夹的名称

是以运行测试用例的日期的方式进行命名。

点入该文件夹，文件夹以测试模块的不同进行命名，即 `test_cli` 和 `test_rpc`，根据想要查看的 log 点入相应的文件夹中。

在该文件夹下，存在两种类型的文件，文件格式为 CSV 的是运行测试用例的汇总表格，log 格式的是每一条运行的 log 文件。如图所示：



Name	Size (KB)	Last modified	Owner
..			
collection_log.csv	2	2019-01-09 09:31	root
test_001_dumpprivkey.log	4	2019-01-09 09:28	root
test_002_dumpprivkey.log	4	2019-01-09 09:28	root
test_003_dumpprivkey.log	4	2019-01-09 09:28	root
test_005_dumpprivkey.log	4	2019-01-09 09:28	root
test_006_dumpprivkey.log	4	2019-01-09 09:28	root
test_007_dumpprivkey.log	4	2019-01-09 09:28	root
test_008_getaccountstate.log	7	2019-01-09 09:28	root
test_009_getaccountstate.log	4	2019-01-09 09:29	root
test_010_getaccountstate.log	4	2019-01-09 09:29	root
test_012_getaccountstate.log	4	2019-01-09 09:29	root
test_013_getaccountstate.log	4	2019-01-09 09:29	root
test_014_getassetstate.log	5	2019-01-09 09:29	root
test_015_getassetstate.log	4	2019-01-09 09:29	root
test_016_getassetstate.log	4	2019-01-09 09:29	root
test_018_getassetstate.log	4	2019-01-09 09:29	root
test_019_getassetstate.log	4	2019-01-09 09:29	root

图 4-3

在 CSV 文件中，每一条运行的 log 会形成一张 CSV 的表格，表格中包含每一条测试用例的方法名，测试结果和测试 log 名称。该表格文件可以清晰明了的反映运行的测试用例运行结果如何，如图所示：

	A	B	C	D	E	F
1	NAME	STATUS	LOG PATH			
2	test_init	block	test_init.log			
3	test_001_dumpprivkey	pass	test_001_dumpprivkey.log			
4	test_002_dumpprivkey	pass	test_002_dumpprivkey.log			
5	test_003_dumpprivkey	pass	test_003_dumpprivkey.log			
6	test_005_dumpprivkey	pass	test_005_dumpprivkey.log			
7	test_006_dumpprivkey	fail	test_006_dumpprivkey.log			
8	test_007_dumpprivkey	pass	test_007_dumpprivkey.log			
9	test_008_getaccountstate	fail	test_008_getaccountstate.log			
10	test_009_getaccountstate	pass	test_009_getaccountstate.log			
11	test_010_getaccountstate	pass	test_010_getaccountstate.log			
12	test_012_getaccountstate	pass	test_012_getaccountstate.log			
13	test_013_getaccountstate	fail	test_013_getaccountstate.log			
14	test_014_getassetstate	pass	test_014_getassetstate.log			
15	test_015_getassetstate	pass	test_015_getassetstate.log			
16	test_016_getassetstate	pass	test_016_getassetstate.log			
17	test_018_getassetstate	pass	test_018_getassetstate.log			
18	test_019_getassetstate	fail	test_019_getassetstate.log			
19	test_020_getbalance	pass	test_020_getbalance.log			
20	test_021_getbalance	fail	test_021_getbalance.log			
21	test_022_getbalance	fail	test_022_getbalance.log			
22	test_024_getbalance	pass	test_024_getbalance.log			
23	test_025_getbalance	fail	test_025_getbalance.log			
24	test_026_getbalance	pass	test_026_getbalance.log			
25	test_027_getbestblockhash	pass	test_027_getbestblockhash.log			
26	test_028_getblock	pass	test_028_getblock.log			

图 4-4

- 7) log 文件夹下存在一个 xlsx 文件,打开文件，第一张 sheet 名为“neo 测试-总计”的，是所有测试用例运行结束后的结果进行的数据统计。

使用方法：

首先点击“清空目前记录”，清空表格中所有以往的记录。

再点击“填入目前所有记录”，在测试结果统计表中，会自动统计每一模块相应的数据，如图所示：

从第 4 张 sheet 开始，是每一个模块的具体测试用例，在“log 信息”一列有每一条 log 信息的链接，点击可以查看，记录的是每一条 log 的具体内容。