



How to use ACME unittests code

ACME unittests code를 사용하는 방법에 대한 문서(Linux)



Table of Contents

I. 가상 환경 설정

II. config.py

III. runTest.py와 test 코드

IV. 실제 사용 화면

CHAPTER

1

가상 환경 설정

1. python 가상 환경 생성 및 실행



- module 설치를 위한 가상 환경

✓ 가상 환경을 설치할 디렉토리를 생성한 후 이동한다.

```
● cumulus@cumulus:~$ mkdir venvs  
● cumulus@cumulus:~$ cd venvs  
○ cumulus@cumulus:~/venvs$
```

✓ 아래 명령어를 사용하여 python 가상 환경을 생성한다.

```
● cumulus@cumulus:~/venvs$ python3 -m venv acme  
● cumulus@cumulus:~/venvs$ ls  
acme  
○ cumulus@cumulus:~/venvs$
```

✓ 그 후 가상환경을 실행하여 모듈을 설치할 준비를 한다.

```
● cumulus@cumulus:~/venvs$ source acme/bin/activate  
○ (acme) cumulus@cumulus:~/venvs$
```

2. ACME에 필요한 모듈 설치



python 모듈 설치

✓ 필요한 모듈을 설치하기 위해 아래 명령어를 입력한다.

```
(acme) cumulus@cumulus:~$ cd ACME-oneM2M-CSE/  
(acme) cumulus@cumulus:~/ACME-oneM2M-CSE$ pip3 install -r requirements.txt
```

✓ 다음 단계를 위하여 “tests” 디렉토리로 이동한다.

```
(acme) cumulus@cumulus:~/ACME-oneM2M-CSE$ cd tests  
(acme) cumulus@cumulus:~/ACME-oneM2M-CSE/tests$ ls  
pycache _ testREQ.py  
config.py testRemote.py  
init.py testRemote_AnnC.py  
mqttRequests testRemote_GRP.py  
runTests.py testRemote_Requests.py  
testACP.py testRequests.py  
testACTR.py testSUB.py  
testAE.py testTS.py  
testAddressing.py testTSB.py  
testCIN.py testTS_TSI.py  
testCNT.py testUpperTester.py  
testCNT_CIN.py  
testFCNT.py  
testFCNT_FCI.py  
testGRP.py  
testLoad.py  
testMgmtObj.py  
testMisc.py  
testNOD.py  
testPCH.py  
testPCH_PCU.py
```

앞으로 사용하게 될
python파일들

CHAPTER

2

`config.py`

1. config.py 코드 구조



● config.py 코드 구조

- ✓ BINDING 변수를 찾아 사용할 프로토콜에 맞추어 설정한다(단, 코드의 버전업에 따라 변수의 위치가 변할 수 있음).

```
10 BINDING = 'http'
11
12 if BINDING == 'mqtt':
13
14     PROTOCOL = 'mqtt'
15     CONFIGPROTOCOL = 'http'
16     NOTIFICATIONPROTOCOL = 'http'
17     REMOTEPROTOCOL = 'http'
18
19 elif BINDING == 'http':
20
21     PROTOCOL = 'http'
22     CONFIGPROTOCOL = 'http'
23     NOTIFICATIONPROTOCOL = 'http'
24     REMOTEPROTOCOL = 'http'
25
26 elif BINDING == 'https':
27
28     PROTOCOL = 'https'
29     CONFIGPROTOCOL = 'https'
30     NOTIFICATIONPROTOCOL = 'http'
31     REMOTEPROTOCOL = 'http'
```

이번 문서에서는
HTTP 프로토콜을
사용할 예정이므로
다음과 같이 기입함

1. config.py 코드 구조



config.py 코드 구조

✔ 그 후 사용하게 될 CSE의 정보, MQTT 관련 정보, OAuth2를 사용할 경우 등에 대한 설정이 나타난다.

<pre> 39 # 40 # CSE Info 41 # 42 43 SERVER = f'{PROTOCOL}://{localhost:8888}' # Remember: no trailing '/' 44 CONFESSESERVER = f'{CONFIGPROTOCOL}://{localhost:8888}' 45 ROOTPATH = '/' 46 CSEIN = '/in-cse/in-name' 47 CSEIS = '/in-cse' 48 CSEID = '/in-cse' 49 SPID = '' 50 ORIGINATOR = 'admin:admin' 51 ORIGINATORResp = 'admin:admin' 52 53 54 # 55 # MQTT (if configured) 56 # 57 58 mqttAddress = 'mqtt' 59 mqttPort = 1883 60 mqttClientId = 'CacoeTest' 61 mqttUsername = 'test' 62 mqttPassword = 'mqtt' 63 64 65 MQTTREQUESTTOPIC = f'{mqttAddress}/req/{ORIGINATORS(CSEID)}/json' 66 MQTTRESPONSETOPIC = f'{mqttAddress}/resp/{ORIGINATORS(CSEID)}/json' 67 MQTTREQUESTTOPIC = f'{mqttAddress}/req/{mqttClientId}/{CSEID}/json' 68 MQTTRESPONSETOPIC = f'{mqttAddress}/resp/{mqttClientId}/{CSEID}/json' 69 70 71 # 72 # OAuth2 authentication 73 # 74 75 doOAuth = False 76 oauthServerUrl = '' 77 oauthClientId = '' 78 oauthClientSecret = '' 79 80 81 # 82 # Remote CSE 83 # 84 85 86 REMOTESERVER = f'{REMOTEPROTOCOL}://{localhost:8881}' 87 REMOTEROOTPATH = '/' 88 REMOTECSEIN = '/cse-in' 89 REMOTECSEIS = '/id-in' 90 REMOTECSEID = '/id-in' 91 REMOTESPID = '/sp-in' 92 REMOTEORIGINATOR = 'Admin' 93 94 95 # 96 # Notification Server 97 # 98 99 NOTIFICATIONPORT = 9999 100 NOTIFICATIONSERVER = f'{NOTIFICATIONPROTOCOL}://{localhost:(NOTIFICATIONPORT)}' 101 NOTIFICATIONSERVER = f'{NOTIFICATIONPROTOCOL}://{localhost:6666}' 102 103 104 # 105 # Upper Tester 106 # 107 108 UTURL = f'{CONFESSESERVER}{ROOTPATH}_ut_' 109 UTCHD = 'X-MQIN-UTCHD' 110 UTNSP = 'X-MQIN-UTNSP' 111 </pre>	CSE 관련 설정
<pre> 54 # 55 # MQTT (if configured) 56 # 57 58 mqttAddress = 'mqtt' 59 mqttPort = 1883 60 mqttClientId = 'CacoeTest' 61 mqttUsername = 'test' 62 mqttPassword = 'mqtt' 63 64 65 MQTTREQUESTTOPIC = f'{mqttAddress}/req/{ORIGINATORS(CSEID)}/json' 66 MQTTRESPONSETOPIC = f'{mqttAddress}/resp/{ORIGINATORS(CSEID)}/json' 67 MQTTREQUESTTOPIC = f'{mqttAddress}/req/{mqttClientId}/{CSEID}/json' 68 MQTTRESPONSETOPIC = f'{mqttAddress}/resp/{mqttClientId}/{CSEID}/json' 69 </pre>	MQTT 관련 설정
<pre> 72 # 73 # OAuth2 authentication 74 # 75 76 doOAuth = False 77 oauthServerUrl = '' 78 oauthClientId = '' 79 oauthClientSecret = '' 80 </pre>	OAuth2 인증 관련 설정
<pre> 82 # 83 # Remote CSE 84 # 85 86 REMOTESERVER = f'{REMOTEPROTOCOL}://{localhost:8881}' 87 REMOTEROOTPATH = '/' 88 REMOTECSEIN = '/cse-in' 89 REMOTECSEIS = '/id-in' 90 REMOTECSEID = '/id-in' 91 REMOTESPID = '/sp-in' 92 REMOTEORIGINATOR = 'Admin' 93 </pre>	Remote CSE 관련 설정
<pre> 95 # 96 # Notification Server 97 # 98 99 NOTIFICATIONPORT = 9999 100 NOTIFICATIONSERVER = f'{NOTIFICATIONPROTOCOL}://{localhost:(NOTIFICATIONPORT)}' 101 NOTIFICATIONSERVER = f'{NOTIFICATIONPROTOCOL}://{localhost:6666}' 102 </pre>	Notification Server 관련 설정
<pre> 104 # 105 # Upper Tester 106 # 107 108 UTURL = f'{CONFESSESERVER}{ROOTPATH}_ut_' 109 UTCHD = 'X-MQIN-UTCHD' 110 UTNSP = 'X-MQIN-UTNSP' 111 </pre>	Upper Tester 관련 설정

2. config.py 설정



• config.py 코드 설정

✓ 이번 테스트를 위한 설정으로 CSE 서버와 관련된 정보만 바꿔주도록 하였다.

이때, 각 정보의 기입란에는 반드시 끝에 '/'를 적지 않도록 한다.

```
39  #
40  #   CSE SuT
41  #
42
43  SERVER          = f'{PROTOCOL}://localhost:8080'    # Remember: no trailing '/'
44  CONFIGSERVER    = f'{CONFIGPROTOCOL}://localhost:8080'
45  ROOTPATH        = '/~'
46  CSERN           = '/in-cse/in-name'
47  CSERI           = '/in-cse'
48  CSEID           = 'in-cse'
49  SPID            = ''
50  ORIGINATOR       = 'admin:admin'
51  ORIGINATORResp   = 'admin:admin'
```

CHAPTER

3

runTest.py와 test 코드

1. runTest.py

● runTest.py란?

- ✓ unittest 코드를 실행하고 결과를 보기 쉽게 정리해주는 코드로 크게 수정할 필요는 없다.

```
(acme) cumulus@cumulus:~/ACME-oneM2M-CSE/tests$ python3 testFCNT_FCI.py
```

```
test_createFCNT (__main__.TestFCNT_FCI)
Create a <FCNT> ... ok
test_attributesFCNT (__main__.TestFCNT_FCI)
Validate <FCNT> attributes ... ok
test_updateFCNT (__main__.TestFCNT_FCI)
Update <FCNT> ... ok
test_retrieveFCNTLaOl (__main__.TestFCNT_FCI)
Retrieve <FCNT> via <FCNT>/la and <FCNT>/ol ... ok
test_updateFCNTMni (__main__.TestFCNT_FCI)
Update <FCNT> MNI ... ok
test_updateLBL (__main__.TestFCNT_FCI)
Update <FCNT> LBL ... ok
test_updateMNIInoFCICreated (__main__.TestFCNT_FCI)
Update MNI, no <FCI> shall be created ... ok
test_createFCIFail (__main__.TestFCNT_FCI)
Create a <FCI> -> Fail ... ok
test_updateFCIFail (__main__.TestFCNT_FCI)
Update a <FCI> -> Fail ... ok
test_updateFCNTMniNull (__main__.TestFCNT_FCI)
Update <FCNT> : set MNI to null ... ok
test_deleteFCNT (__main__.TestFCNT_FCI)
Delete <FCNT> ... ERROR
ERROR
```

```
ERROR: test_deleteFCNT (__main__.TestFCNT_FCI)
Delete <FCNT>
-----
Traceback (most recent call last):
  File "testFCNT_FCI.py", line 242, in test_deleteFCNT
    rsc = DELETE(fcURL, ORIGINATOR)
  File "/home/cumulus/ACME-oneM2M-CSE/tests/init.py", line 285, in DELETE
    x, rsc = sendRequest(Operation.DELETE, url, originator, headers=headers)
  File "/home/cumulus/ACME-oneM2M-CSE/tests/init.py", line 302, in sendRequest
    return sendHttpRequest(requests.delete, url=url, originator=originator, ty=ty, data=data, ct=ct, timeout=timeout, headers=headers)
  File "/home/cumulus/ACME-oneM2M-CSE/tests/init.py", line 387, in sendHttpRequest
    elif ct.startswith(('application/json', 'application/vnd.onem2m-res+json')):
AttributeError: 'NoneType' object has no attribute 'startswith'
-----
ERROR: tearDownClass (__main__.TestFCNT_FCI)
-----
Traceback (most recent call last):
  File "testFCNT_FCI.py", line 43, in tearDownClass
    DELETE(urlURL, ORIGINATOR) # Just delete the AE and everything below it. Ignore whether it exists or not
  File "/home/cumulus/ACME-oneM2M-CSE/tests/init.py", line 285, in DELETE
    x, rsc = sendRequest(Operation.DELETE, url, originator, headers=headers)
  File "/home/cumulus/ACME-oneM2M-CSE/tests/init.py", line 302, in sendRequest
    return sendHttpRequest(requests.delete, url=url, originator=originator, ty=ty, data=data, ct=ct, timeout=timeout, headers=headers)
  File "/home/cumulus/ACME-oneM2M-CSE/tests/init.py", line 387, in sendHttpRequest
    elif ct.startswith(('application/json', 'application/vnd.onem2m-res+json')):
AttributeError: 'NoneType' object has no attribute 'startswith'
-----
Ran 11 tests in 0.841s

FAILED (errors=2)
(acme) cumulus@cumulus:~/ACME-oneM2M-CSE/tests$
```

runTest.py를 사용하지 않은 경우

```
● (acme) cumulus@cumulus:~/ACME-oneM2M-CSE/tests$ python3 runTests.py testFCNT_FCI.py
```

```
Ran 11 tests in 0.933s
```

```
FAILED (errors=2)
```

```
Successfully executed tests: 11
```

```
Errors: 2
```

runTest.py를 사용한 경우

[ACME] - Test Results

Test Suite	Count	Skipped	Errors	Exec Time	Process Time	Time / Test	Requests
testFCNT_FCI	11	0	2	0.9338	0.0922	0.0849	23
Totals	11	0	2	0.9440	0.1016	0.0858	23

```
(acme) cumulus@cumulus:~/ACME-oneM2M-CSE/tests$
```

2. test 코드

- testFCNT_FCI.py

- ✓ 이 문서에서는 FlexContainer의 FlexContainerInstance가 올바르게 구현되었는지에 대한 unittest를 실시한다.
- ✓ 그러나 모든 unittest 코드들은 ACME를 기준으로 작성되었기 때문에 직접 개발한 oneM2M 플랫폼에는 올바르게 동작할 가능성이 존재한다.
- ✓ 때문에 test 코드의 간단한 구조와 디버깅 시 자신의 oneM2M 플랫폼에 맞출 수 있는 방법을 설명하겠다.

2. test 코드

• testFCNT_FCI.py

- ✓ main 코드를 보면 run 함수를 실행하는 것을 알 수 있고 run 함수에 어떤 테스트를 실시할 것인지 TestFCNT_FCI의 인자를 보고 직관적으로 알 수 있다.

```
246 # TODO other FCNT controlling attributes
247 # TODO Add similar tests from testCNT_CIN for mni, etc
248
249 def run(testVerbosity:int, testFailFast:bool) -> Tuple[int, int, int]:
250     suite = unittest.TestSuite()
251     suite.addTest(TestFCNT_FCI('test_createFCNT'))
252     suite.addTest(TestFCNT_FCI('test_attributesFCNT'))
253     suite.addTest(TestFCNT_FCI('test_updateFCNT'))
254     suite.addTest(TestFCNT_FCI('test_retrieveFCNTLa01'))
255     suite.addTest(TestFCNT_FCI('test_updateFCNTMni'))
256     suite.addTest(TestFCNT_FCI('test_updateLBL'))
257     suite.addTest(TestFCNT_FCI('test_updateMNInoFCICreated'))
258     suite.addTest(TestFCNT_FCI('test_createFCIFail'))
259     suite.addTest(TestFCNT_FCI('test_updateFCIFail'))
260     suite.addTest(TestFCNT_FCI('test_updateFCNTMniNull'))
261     suite.addTest(TestFCNT_FCI('test_deleteFCNT'))
262     result = unittest.TextTestRunner(verbosity=testVerbosity, failfast=testFailFast).run(suite)
263     printResult(result)
264     return result.testsRun, len(result.errors + result.failures), len(result.skipped)
265
266
267 if __name__ == '__main__':
268     _, errors, _ = run(2, True)
269     sys.exit(errors)
270
271
```

2. test 코드

● testFCNT_FCI.py

- ✓ 전달한 인자를 가지고 수행할 함수를 찾게 된다.

예를 들어 'test_createFCNT' 함수를 살펴보면 아래와 같다.

- ✓ 테스트 통과 유무는 붉은색 표시가 된 self.assert[조건]에서 조건이 참이 되면 테스트를 통과하게 된다.

test를 위해 사용될 AE 리소스가 생성되었는지 확인하는 테스트, 생성된 상태라면 테스트를 통과함

```

46 @unittest.skipIf(noCSE, 'No CSEBase')
47 def test_createFCNT(self) -> None:
48     """ Create a <FCNT> """
49     self.assertIsNotNone(TestFCNT_FCI.ae)
50     dct = { 'hd:tempe' : {
51             'rn' : fcntRN,
52             'cnd' : CND,
53             'curT0' : 23.0,
54             'unit' : 1,
55             'minVe' : -100.0,
56             'maxVe' : 100.0,
57             'steVe' : 0.5,
58             'mni' : 10
59         }}
60     r, rsc = CREATE(aeURL, ORIGINATOR, T.FCNT, dct)
61     self.assertEqual(rsc, RC.created)

```

FlexContainer CREATE request 송신 후 전달받은 response에 CREATE 후 받을 수 있는 올바른 statusCode가 있는지 확인하는 테스트, 2001을 전달받으면 테스트를 통과함

2. test 코드

● testFCNT_FCI.py

- ✔ 만약 테스트를 수행할 때 oneM2M 표준 문서에서 설명한대로 잘 구현했지만, 테스트 통과 Error가 발생할 경우, 이것이 ACME에 의존적인 부분이라 판단된다면 해당 부분을 주석 처리하거나 따로 수정하여 테스트를 진행하면 됨

```

64 @unittest.skipIf(noCSE, 'No CSEBase')
65 def test_attributesFCNT(self) -> None:
66     """ Validate <FCNT> attributes """
67     r, rsc = RETRIEVE(fcncURL, ORIGINATOR)
68     self.assertEqual(rsc, RC.OK, r)
69     self.assertEqual(findXPath(r, 'hd:tempe/ty'), T.FCNT, r)
70     self.assertEqual(findXPath(r, 'hd:tempe/pi'), findXPath(TestFCNT_FCI.ae, 'm2m:ae/ri'), r)
71     self.assertEqual(findXPath(r, 'hd:tempe/rn'), fcncRN, r)
72     self.assertIsNotNone(findXPath(r, 'hd:tempe/ct'), r)
73     self.assertIsNotNone(findXPath(r, 'hd:tempe/lt'), r)
74     self.assertIsNotNone(findXPath(r, 'hd:tempe/et'), r)
75     #self.assertIsNotNone(findXPath(r, 'hd:tempe/st'), r)
76     self.assertIsNone(findXPath(r, 'hd:tempe/cr'), r)
77     self.assertEqual(findXPath(r, 'hd:tempe/cnd'), CND, r)
78     self.assertEqual(findXPath(r, 'hd:tempe/curID'), '23.0', r)
79     ##self.assertIsNone(findXPath(r, 'hd:tempe/tarTe'), r)
80     self.assertEqual(findXPath(r, 'hd:tempe/unit'), '1', r)
81     self.assertEqual(findXPath(r, 'hd:tempe/minVe'), '-100.0', r)
82     self.assertEqual(findXPath(r, 'hd:tempe/maxVe'), '100.0', r)
83     self.assertEqual(findXPath(r, 'hd:tempe/stVe'), '0.5', r)
84     #self.assertIsNotNone(findXPath(r, 'hd:tempe/st'), r)
85     #self.assertEqual(findXPath(r, 'hd:tempe/st'), 0, r)
86     self.assertIsNotNone(findXPath(r, 'hd:tempe/mni'), r)
87     self.assertEqual(findXPath(r, 'hd:tempe/mni'), 10, r)
88     self.assertIsNotNone(findXPath(r, 'hd:tempe/cni'), r)
89     self.assertEqual(findXPath(r, 'hd:tempe/cni'), 1, r)
90     self.assertIsNotNone(findXPath(r, 'hd:tempe/cbs'), r)
91     #self.assertGreater(findXPath(r, 'hd:tempe/cbs'), 0, r)

```

단순히 예시를 위한 코드로
주석 처리한 코드가 실제로
ACME에 의존적인 부분은
아님을 밝힘

CHAPTER

4

실제 사용 화면

1. 결과

● 결과

- ✓ 11개의 테스트가 성공했고 2개의 Error 상황이 발생하였다.
- ✓ 이때 유의할 점은 이 test code는 ACME의 구현에 의존하여 개발되었기 때문에 Error를 출력했다고 하여 반드시 개발한 oneM2M 플랫폼이 잘못 구현된 것은 아니다.

```
-----
Ran 11 tests in 0.933s

FAILED (errors=2)
Successfully executed tests: 11
Errors: 2
```

Error 상황에 대한 디버그는 상단에 나타난 정보들을 기반으로 수정해 나가면 됨.

```

[ACME] - Test Results

```

Test Suite	Count	Skipped	Errors	Exec Time	Process Time	Time / Test	Requests
testFCNT_FCI	11	0	2	0.9338	0.0922	0.0849	23
Totals	11	0	2	0.9440	0.1016	0.0858	23

```

(acme) cumulus@cumulus:~/ACME-oneM2M-CSE/tests$

```



THANK
YOU!