1. Тестирование

1.1. Модульное тестирование

Производится модульное тестирование модуля сервера данных, отвечающего за работу комплексного запроса get_free_subtask_by_agent_id.

Тестрирование производится при помощи библиотеки unittest в автоматическом режиме. Тест самостоятельно геренирует входные данные. Тест не генерирует дополнительного вывода в случае корректной работы, кроме унифицированного библиотекой unittest.

Код теста

```
#!/bin/env python
import sys
import os

PACKAGE_PARENT = '..'
SCRIPT_DIR = os.path.dirname(os.path.realpath(os.path.join(os.getcwd(), os.path.
    expanduser(__file__))))
sys.path.append(os.path.normpath(os.path.join(SCRIPT_DIR, PACKAGE_PARENT)))

import unittest
from data_backend import post_item, get_free_subtask_by_agent_id, init_db

class utDataBackend(unittest.TestCase):

    def test_get_free_subtask_by_agent_id(self):
        def ut_assert(self, qry, code = 200):
            self.assertTrue(qry == code)
        init_db()

        ut_assert(self, post_item('trait', {"name":"trait1","version":"1.0"}))
        ut_assert(self, post_item('trait', {"name":"trait2","version":"2.0"}))
        ut_assert(self, post_item('trait', {"name":"trait3","version":"3.0"}))
        ut_assert(self, post_item('trait', {"name":"trait4","version":"4.0"}))

        ut_assert(self, post_item('agent', {}))
        ut_assert(self, post_item('agent', {}))

        ut_assert(self, post_item('task', {"max_time":160,"archive_name":"cocoque"
            }))
```

```
28          ut_assert( self , post_item('task', {"max_time":260,"archive_name":"
                kukareque"}))
29
30          ut_assert( self , post_item('subtask', {"task_id":1,"status":"queued","
                archive_name":"cocoque0"}))
31          ut_assert( self , post_item('subtask', {"task_id":2,"status":"queued","
                archive_name":"cocoque1"}))
32          ut_assert( self , post_item('subtask', {"task_id":1,"status":"queued","
                archive_name":"cocoque2"}))
33          ut_assert( self , post_item('subtask', {"task_id":2,"status":"queued","
                archive_name":"cocoque3"}))
34          ut_assert( self , post_item('subtask', {"task_id":1,"status":"queued","
                archive_name":"cocoque4"}))
35
36          ut_assert( self , post_item('mtm_traitagent', {"trait_id":1,"agent_id":2}))
37          ut_assert( self , post_item('mtm_traitagent', {"trait_id":2,"agent_id":3}))
38          ut_assert( self , post_item('mtm_traitagent', {"trait_id":3,"agent_id":4}))
39          ut_assert( self , post_item('mtm_traitagent', {"trait_id":4,"agent_id":1}))
40          ut_assert( self , post_item('mtm_traitagent', {"trait_id":2,"agent_id":2}))
41          ut_assert( self , post_item('mtm_traitagent', {"trait_id":4,"agent_id":4}))
42
43          ut_assert( self , post_item('mtm_traittask', {"trait_id":1,"task_id":1}))
44          ut_assert( self , post_item('mtm_traittask', {"trait_id":2,"task_id":1}))
45          ut_assert( self , post_item('mtm_traittask', {"trait_id":3,"task_id":1}))
46          ut_assert( self , post_item('mtm_traittask', {"trait_id":2,"task_id":2}))
47          ut_assert( self , post_item('mtm_traittask', {"trait_id":4,"task_id":2}))
48          ut_assert( self , post_item('mtm_traittask', {"trait_id":1,"task_id":2}))
49
50          ut_assert( self , get_free_subtask_by_agent_id(1))
51          ut_assert( self , get_free_subtask_by_agent_id(1))
52          ut_assert( self , get_free_subtask_by_agent_id(1), 404)
53          ut_assert( self , get_free_subtask_by_agent_id(2))
54          ut_assert( self , get_free_subtask_by_agent_id(2))
55          ut_assert( self , get_free_subtask_by_agent_id(2))
56          ut_assert( self , get_free_subtask_by_agent_id(2), 404)
57
58          ut_assert( self , get_free_subtask_by_agent_id(10), 404)
59
60          ut_assert( self , get_free_subtask_by_agent_id('lkkl'), 400)
61
62  if __name__ == '__main__':
63      unittest.main()
```

Унифицированный вывод библиотеки unittest

```
----------------------------------------------------------------------

Ran 1 test in 1.231s

OK
```

1.2. Системное тестирование

Производится системное тестирования файлового сервера.

Тест покрывает 100% АПИ файлового сервера. Тест самостоятельно генерирует входные данные.

Код теста

```
#!/bin/bash

echo "#include <iostream>" > test.cpp
echo "int main(){std::cout << \"HELLO WORLD!\" << std::endl;}" >> test.cpp
g++ test.cpp -o a.out

set -v

curl @"localhost:50002/static?path=1\2\3\4" -X POST -F file=@a.out; echo
curl @"localhost:50002/static"              -X POST -F file=@a.out; echo

curl @"localhost:50002/static/1\2\3\4\a.out" -X GET > a_.out     ; echo
chmod +x a_.out                                                  ; echo
./a_.out                                                         ; echo

curl @"localhost:50002/static/a.out"         -X GET > a__.out    ; echo
chmod +x a__.out                                                 ; echo
./a__.out                                                        ; echo

curl @"localhost:50002/static/1\2\3\4\a.out" -X DELETE           ; echo
curl @"localhost:50002/static/a.out"         -X DELETE           ; echo

curl @"localhost:50002/static/1\2\3\4\a.out" -X GET              ; echo
curl @"localhost:50002/static/a.out"         -X GET              ; echo

rm a*.out test.cpp
```

Выходные данные

```
curl @"localhost:50002/static?path=1\2\3\4" -X POST -F file=@a.out; echo
{}
curl @"localhost:50002/static"              -X POST -F file=@a.out; echo
{}

curl @"localhost:50002/static/1\2\3\4\a.out" -X GET > a_.out     ; echo
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  8384  100  8384    0     0  4885k       0 --:--:-- --:--:-- --:--:-- 8187
     k

chmod +x a_.out                                                  ; echo

./a_.out                                                         ; echo
HELLO WORLD!


curl @"localhost:50002/static/a.out"        -X GET > a__.out    ; echo
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  8384  100  8384    0     0  4047k       0 --:--:-- --:--:-- --:--:-- 8187
     k

chmod +x a__.out                                                 ; echo

./a__.out                                                        ; echo
HELLO WORLD!


curl @"localhost:50002/static/1\2\3\4\a.out" -X DELETE           ; echo
{}
curl @"localhost:50002/static/a.out"        -X DELETE           ; echo
{}

curl @"localhost:50002/static/1\2\3\4\a.out" -X GET              ; echo
{
  "error": "Not found"
}
curl @"localhost:50002/static/a.out"        -X GET              ; echo
{
  "error": "Not found"
}

```

```
42 rm a*.out test.cpp
```

1.3. Интеграционное тестирование

Производится интеграционное тестирование подсистем работы с файлами, данными и подсистемы мониторинга.

Тест покрывает прецедент регистрации подсистем работы с файлами и данными в подсистеме мониторинга. Тест не требует входных данных.

Код теста

```
1 #!/bin/bash
2 python ../beacon_backend/beacon_backend.py 50003 &
3 python ../data_backend/data_backend.py localhost:50003 10.0.0.10:5432 50001 &
4 python ../file_backend/file_backend.py localhost:50003 50002 &
5
6 echo "sleep 30"
7 sleep 50
8
9 killall -9 python
```

Выходные данные

```
1  * Running on http://0.0.0.0:50003/ (Press CTRL+C to quit)
2  * Restarting with stat
3 Starting with settings: beacon: localhost:50003 self: 0.0.0.0:50002
4 Starting with settings: Beacon: localhost:50003 DB: 10.0.0.10:5432, self:
      0.0.0.0:50001
5 Beacon is down. Waiting to reconnect.
6 Incoming request from 127.0.0.1: port = 50003, state = Unable to find beacon
7 127.0.0.1 - - [28/May/2015 02:23:31] "POST /services/fileserver HTTP/1.1" 200 -
8 Beacon is back up.
9  * Running on http://0.0.0.0:50002/ (Press CTRL+C to quit)
10  * Restarting with stat
11 Incoming request from 127.0.0.1: port = 50003, state = Operating normally
12 127.0.0.1 - - [28/May/2015 02:23:31] "POST /services/database HTTP/1.1" 200 -
13  * Running on http://0.0.0.0:50001/ (Press CTRL+C to quit)
14  * Restarting with stat
15 Starting with settings: beacon: localhost:50003 self: 0.0.0.0:50002
16 Incoming request from 127.0.0.1: port = 50003, state = Operating normally
17 127.0.0.1 - - [28/May/2015 02:23:32] "POST /services/fileserver HTTP/1.1" 200 -
18 Starting with settings: Beacon: localhost:50003 DB: 10.0.0.10:5432, self:
      0.0.0.0:50001
19 Incoming request from 127.0.0.1: port = 50003, state = Operating normally
```

```
20 127.0.0.1 - - [28/May/2015 02:23:32] "POST /services/database HTTP/1.1" 200 -
21 Incoming request from 127.0.0.1:50003: state = Operating normally
22 127.0.0.1 - - [28/May/2015 02:23:41] "PUT /services/fileserver/127.0.0.1:50003
       HTTP/1.1" 200 -
23 Incoming request from 127.0.0.1:50003: state = Operating normally
24 127.0.0.1 - - [28/May/2015 02:23:41] "PUT /services/database/127.0.0.1:50003
       HTTP/1.1" 200 -
25 Incoming request from 127.0.0.1:50003: state = Operating normally
26 127.0.0.1 - - [28/May/2015 02:23:42] "PUT /services/fileserver/127.0.0.1:50003
       HTTP/1.1" 200 -
27 Incoming request from 127.0.0.1:50003: state = Operating normally
28 127.0.0.1 - - [28/May/2015 02:23:42] "PUT /services/database/127.0.0.1:50003
       HTTP/1.1" 200 -
29 Incoming request from 127.0.0.1:50003: state = Operating normally
30 127.0.0.1 - - [28/May/2015 02:23:51] "PUT /services/fileserver/127.0.0.1:50003
       HTTP/1.1" 200 -
31 Incoming request from 127.0.0.1:50003: state = Operating normally
32 127.0.0.1 - - [28/May/2015 02:23:51] "PUT /services/database/127.0.0.1:50003
       HTTP/1.1" 200 -
33 Incoming request from 127.0.0.1:50003: state = Operating normally
34 127.0.0.1 - - [28/May/2015 02:23:52] "PUT /services/fileserver/127.0.0.1:50003
       HTTP/1.1" 200 -
35 Incoming request from 127.0.0.1:50003: state = Operating normally
36 127.0.0.1 - - [28/May/2015 02:23:52] "PUT /services/database/127.0.0.1:50003
       HTTP/1.1" 200 -
37 Incoming request from 127.0.0.1:50003: state = Operating normally
38 127.0.0.1 - - [28/May/2015 02:24:01] "PUT /services/fileserver/127.0.0.1:50003
       HTTP/1.1" 200 -
39 Incoming request from 127.0.0.1:50003: state = Operating normally
40 127.0.0.1 - - [28/May/2015 02:24:01] "PUT /services/database/127.0.0.1:50003
       HTTP/1.1" 200 -
41 Incoming request from 127.0.0.1:50003: state = Operating normally
42 127.0.0.1 - - [28/May/2015 02:24:02] "PUT /services/fileserver/127.0.0.1:50003
       HTTP/1.1" 200 -
43 Incoming request from 127.0.0.1:50003: state = Operating normally
44 127.0.0.1 - - [28/May/2015 02:24:02] "PUT /services/database/127.0.0.1:50003
       HTTP/1.1" 200 -
45 Incoming request from 127.0.0.1:50003: state = Operating normally
46 127.0.0.1 - - [28/May/2015 02:24:11] "PUT /services/fileserver/127.0.0.1:50003
       HTTP/1.1" 200 -
47 Incoming request from 127.0.0.1:50003: state = Operating normally
48 127.0.0.1 - - [28/May/2015 02:24:11] "PUT /services/database/127.0.0.1:50003
       HTTP/1.1" 200 -
```

```
49 Incoming request from 127.0.0.1:50003: state = Operating normally
50 127.0.0.1 - - [28/May/2015 02:24:12] "PUT /services/fileserver/127.0.0.1:50003
       HTTP/1.1" 200 -
51 Incoming request from 127.0.0.1:50003: state = Operating normally
52 127.0.0.1 - - [28/May/2015 02:24:12] "PUT /services/database/127.0.0.1:50003
       HTTP/1.1" 200 -
```