

branch-api короткий опис та приклади unit тестів

Цей API розроблено для того, щоб завести під корпорацію branch, що фактично буде робити виплати на карти

Методи API

baseURL: /branch-api/branch

http-put - створити branch

► Детально про метод

- обов'язкові http заголовки

```
Content-Type: application/json
```

- параметри в path передається corporateid **/branch-api/branch/:corporateid**

http://localhost:8080/branch-api/branch/CORP-68f6002dfa4e1d8d2711b8fc03bf8c94

- запит

```
{
  "type": "BRANCH",
  "branch_name": "Харківська філія",
  "branch_erd pou": "14320930",
  "branch_adress": "м. Харків",
  "branch_phone": "(066) 224-19-78",
  "branch_sign_certs": [
    "cert-ffffff",
    "cert-dddddddddddd",
    "cert-TTTTTTTTTTTTTT"
  ],
  "branch_encrypt_certs": [
    "cert-vvvvvvvvvvvvvv"
  ]
}
```

- відповідь

```
{"branchid": "BRANCH-681099c6c2b601b63fa1746c73edc789"}
```

http-get - (без параметрів) отримати список всіх бренчів по корпорації

► Детально про метод

- обов'язкові http заголовки

```
Content-Type: application/json
```

- відповідь

```
[{
  "_id": "BRANCH-49a66d8714b1f711a80e44fc2d0da0af",
  "_rev": "1-fe18aeb5737f30dead11421911820e95",
  "corporateid": "CORP-68f6002dfa4e1d8d2711b8fc03bf8c94",
  "type": "BRANCH",
  "branch_name": "Полтавська філія",
  "branch_erd pou": "14360940",
  "branch_adress": "В Полтаві десть, біля ярмарк",
  "branch_phone": "(044) 2241416",
  "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-
TTTTTTTTTTTTTT"],
  "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]
}, {
  "_id": "BRANCH-93ff6628157b0afa26fbfb1463dbd136",
  "_rev": "1-c03a6130cda501c77095f57c1223843f",
  "corporateid": "CORP-68f6002dfa4e1d8d2711b8fc03bf8c94",
  "type": "BRANCH",
  "branch_name": "Jason Wolfe",
  "branch_erd pou": "zademwulubvik",
  "branch_adress": "sitowituebrufeu",
  "branch_phone": "(776) 224-1978",
  "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-
TTTTTTTTTTTTTT"],
  "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]
}, {
  "_id": "BRANCH-e468d381eef4ecf020692ec89a7b9b4e",
  "_rev": "1-ecf6fb71462799bd03bd288e300abe45",
  "corporateid": "CORP-68f6002dfa4e1d8d2711b8fc03bf8c94",
  "type": "BRANCH",
  "branch_name": "Дніпровськ філія",
  "branch_erd pou": "14360930",
  "branch_adress": "У Дніпрв",
  "branch_phone": "(776) 224-1978",
  "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-
TTTTTTTTTTTTTT"],
  "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]
}]
```

http-get - (з параметром id корпорації та branchid) отримати дані одного брэнча корпорації

► Детально про метод

- обов'язкові http заголовки

```
Content-Type: application/json
```

- запит

В URL передається ідентифікатор корпорації та бренча: /branch-api/branch/:corporateid/:branchid

url: http://localhost:8080/branch-api/branch/CORP-681099c6c2b601b63fa1746c73edc789/BRANCH-93ff6628157b0afa26fbfb1463dbd136

- відповідь

```
{
  "_id": "BRANCH-e468d381eef4ecf020692ec89a7b9b4e",
  "_rev": "1-ecf6fb71462799bd03bd288e300abe45",
  "corporateid": "CORP-68f6002dfa4e1d8d2711b8fc03bf8c94",
  "type": "BRANCH",
  "branch_name": "Дніпровськ філія",
  "branch_erdpo": "14360930",
  "branch_adress": "У Дніпрв",
  "branch_phone": "(776) 224-1978",
  "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-
TTTTTTTTTTTTTTTT"],
  "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]
}
```

http-delete - (з параметром id корпорації та id branch) видалити дані одного бренча по корпорації

► Детально про метод

- обов'язкові http заголовки

```
Content-Type: application/json
```

- запит

В URL передається ідентифікатор корпорації та branch: /branch-api/branch/:corporateid/:branchid

url: http://localhost:8080/branch-api/branch/CORP-681099c6c2b601b63fa1746c73edc789/BRANCH-93ff6628157b0afa26fbfb1463dbd136

- відповідь

```
{ "id": "BRANCH-681099c6c2b601b63fa1746c73edc789", "rev": "2-34292cda93dd4d6b3fd334ac0ee45bb9", "ok": true }
```

Unit-тести

Для тестів створено стандартний тестовий клас: **TestBranchAPI**, з наступними функціями

- test_createBranch - створити branch
- test_getBranchList - отримати список всіх бранчів корпорації
- test_getBranchItem - отримати реквізити одного бранча корпорації
- test_deleteBranchItem - видалити зі списку один бранч корпорації

в тестах перевіряється тільки відповідність відповіді на http status code =200

```
import requests
import unittest
import sys
import logging
import json
import datetime
import base64

logging.basicConfig(stream=sys.stdout, level=logging.INFO)

class TestBranchAPI(unittest.TestCase):
    baseurl="http://localhost:8080"
    requests={}
    branches={}
    requests["corporateid"]="CORP-821d5655a7dcd7f1017eaae1db60ffe6"
    requests["createBranch"]={
        "branch_name": "Факультет Информатик",
        "branch_erd pou": "14360931",
        "branch_adress": "м.Київ",
        "branch_phone": "(050)-222-33-44",
        "branch_sign_certs": ["cert-ffffff", "cert-
ddddd", "cert-TTTTTTTTTTTTTT"],
        "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]
    }

    """
    requests["createBranch"]={
        "branch_name": "Факультет біології",
        "branch_erd pou": "14370940",
        "branch_adress": "м.Київ",
        "branch_phone": "(050)-333-22-11",
        "branch_sign_certs": ["cert-ffffff", "cert-
ddddd", "cert-TTTTTTTTTTTTTT"],
        "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]
    }
```

```

"""

def setUp(self):
    log=logging.getLogger('setUp')
    #log.info("1")

def tearDown(self):
    log=logging.getLogger('tearDown')
    #log.info("2")

def test_createBranch(self):
    """===== Створити branch по корпорації ====="""
    log=logging.getLogger('test_createBranch')
    log.info( f"Опис тесту: {self.shortDescription()} " )

    log.info( f"Отримую тестовий запит")
    req_obj = TestBranchAPI.requests["createBranch"]
    req_url=TestBranchAPI.baseurl + f"/branch-
api/branch/{TestBranchAPI.requests['corporateid']}"
    req_data=json.dumps(req_obj, ensure_ascii=False )
    log.info(f"Формую запит на url: {req_url}")
    log.info("Method: http-PUT")
    log.info(f"Тіло запиту: {req_data}")

    log.info("Виконю http запит")
    response = requests.put(req_url, data=req_data , headers={'Content-Type':
'application/json'} )
    log.info(f"Response: {response.text}")
    log.info("Перевіря відповідь statusCode=200")
    self.assertEqual(response.status_code, 200)
    res_dict=response.json()

def test_getBranchList(self):
    """===== Отримати список всіх брэнчів корпорації
===== """
    log=logging.getLogger('test_getBranchList')
    log.info( f"Опис тесту: {self.shortDescription()} " )
    req_url=TestBranchAPI.baseurl + f"/branch-
api/branch/{TestBranchAPI.requests['corporateid']}"

    log.info(f"Формую запит на url: {req_url}")
    log.info("Method: http-get")
    log.info("Виконю http запит")
    response = requests.get(req_url, headers={'Content-Type':
'application/json'} )
    log.info(f"Response: {response.text}")
    log.info("Перевіря відповідь statusCode=200")
    self.assertEqual(response.status_code, 200)
    res_dict=response.json()
    TestBranchAPI.branches["list"]=res_dict

```

```
def test_getBranchItem(self):
    """===== Отримати одну корпорацію оп id
    ====="""
    log=logging.getLogger('test_getBranchItem')
    log.info( f"Опис тесту: {self.shortDescription()}")
    branchList=TestBranchAPI.branches["list"]
    if len(branchList) > 0:
        branchid=branchList[0]["_id"]
        req_url=TestBranchAPI.baseurl + f"/branch-
api/branch/{TestBranchAPI.requests['corporateid']}/{branchid}"
        log.info(f"Формую запит на url: {req_url}")
        log.info("Method: http-get")
        log.info("Виконю http запит")
        response = requests.get(req_url, headers={'Content-Type':
'application/json'})
        log.info(f"Response: {response.text}")
        log.info("Перевіря відповідь statusCode=200")
        self.assertEqual(response.status_code, 200)
        res_dict=response.json()

def test_deleteBranchItem(self):
    """===== Видалити один бренср в корпорації оп id
    ====="""
    log=logging.getLogger('test_deleteBranchItem')
    log.info( f"Опис тесту: {self.shortDescription()}")
    branchList=TestBranchAPI.branches["list"]
    if len(branchList) > 0:
        branchid=branchList[0]["_id"]
        req_url=TestBranchAPI.baseurl + f"/branch-
api/branch/{TestBranchAPI.requests['corporateid']}/{branchid}"
        log.info(f"Формую запит на url: {req_url}")
        log.info("Method: http-get")
        log.info("Виконю http запит")
        response = requests.delete(req_url, headers={'Content-Type':
'application/json'})
        log.info(f"Response: {response.text}")
        log.info("Перевіря відповідь statusCode=200")
        self.assertEqual(response.status_code, 200)
        res_dict=response.json()

def suite():
    suite = unittest.TestSuite()
    suite.addTest(TestBranchAPI('test_createBranch'))
    suite.addTest(TestBranchAPI('test_getBranchList'))
    suite.addTest(TestBranchAPI('test_getBranchItem'))
    suite.addTest(TestBranchAPI('test_deleteBranchItem'))

    return suite

if __name__ == '__main__':
    runner = unittest.TextTestRunner()
    runner.run(suite())
```

INFO:test_createBranch:Опис тесту: ===== Створити branch по корпорації
=====

INFO:test_createBranch:Отримую тестовий запит

INFO:test_createBranch:Формую запит на url: http://localhost:8080/branch-api/branch/CORP-821d5655a7dcd7f1017eaae1db60ffe6

INFO:test_createBranch:Method: http-PUT

INFO:test_createBranch:Тіло запиту: {"branch_name": "Факультет Інформатик", "branch_erdpou": "14360931", "branch_adress": "м.Київ", "branch_phone": "(050)-222-33-44", "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-TTTTTTTTTTTTTT"], "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]}

INFO:test_createBranch:Виконю http запит

INFO:test_createBranch:Response: {"branchid": "BRANCH-639dc0786e89cc89eb100f22fd0b24ab"}

INFO:test_createBranch:Перевіря відповідь statusCode=200

.

INFO:test_getBranchList:Опис тесту: ===== Отримати список всіх бренчів корпорації =====

INFO:test_getBranchList:Формую запит на url: http://localhost:8080/branch-api/branch/CORP-821d5655a7dcd7f1017eaae1db60ffe6

INFO:test_getBranchList:Method: http-get

INFO:test_getBranchList:Виконю http запит

INFO:test_getBranchList:Response: [{"_id": "BRANCH-3c94f4eaa5ece90467159915a0ba081f", "_rev": "1-450fe7aa2630817216731f8a7dcd3387", "corporateid": "CORP-821d5655a7dcd7f1017eaae1db60ffe6", "type": "BRANCH", "branch_name": "Факультет Інформатик", "branch_adress": "м.Київ", "branch_phone": "(050)-222-33-44", "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-TTTTTTTTTTTTTT"], "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]}, {"_id": "BRANCH-639dc0786e89cc89eb100f22fd0b24ab", "_rev": "1-450fe7aa2630817216731f8a7dcd3387", "corporateid": "CORP-821d5655a7dcd7f1017eaae1db60ffe6", "type": "BRANCH", "branch_name": "Факультет Інформатик", "branch_adress": "м.Київ", "branch_phone": "(050)-222-33-44", "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-TTTTTTTTTTTTTT"], "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]}, {"_id": "BRANCH-6575940259a50407546737e6843b92fc", "_rev": "1-d610eab846980187e5f0f3d143f74b02", "corporateid": "CORP-821d5655a7dcd7f1017eaae1db60ffe6", "type": "BRANCH", "branch_name": "Факультет біології", "branch_adress": "м.Ктїв", "branch_phone": "(050)-333-22-11", "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-TTTTTTTTTTTTTT"], "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]}, {"_id": "BRANCH-8889189113b8a815cd8538c09a96fea8", "_rev": "1-450fe7aa2630817216731f8a7dcd3387", "corporateid": "CORP-821d5655a7dcd7f1017eaae1db60ffe6", "type": "BRANCH", "branch_name": "Факультет Інформатик", "branch_adress": "м.Київ", "branch_phone": "(050)-222-33-44", "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-TTTTTTTTTTTTTT"], "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvvvv"]}, {"_id": "BRANCH-

```
e279b6c7c6fdde3b0aa97d5dcd6605a", "_rev": "1-450fe7aa2630817216731f8a7dcd3387", "corporateid": "CORP-821d5655a7dcd7f1017eaae1db60ffe6", "type": "BRANCH", "branch_name": "Факультет Інформатик", "branch_adress": "м.Київ", "branch_phone": "(050)-222-33-44", "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-TTTTTTTTTTTTTT"], "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvv"]}]
INFO:test_getBranchList:Перевіря відповідь statusCode=200
```

.

```
INFO:test_getBranchItem:Опис тесту: ===== Отримати одну корпорацію on id =====
```

```
INFO:test_getBranchItem:Формую запит на url: http://localhost:8080/branch-api/branch/CORP-821d5655a7dcd7f1017eaae1db60ffe6/BRANCH-3c94f4eaa5ece90467159915a0ba081f
```

```
INFO:test_getBranchItem:Method: http-get
```

```
INFO:test_getBranchItem:Виконю http запит
```

```
INFO:test_getBranchItem:Response: {"_id": "BRANCH-
```

```
3c94f4eaa5ece90467159915a0ba081f", "_rev": "1-450fe7aa2630817216731f8a7dcd3387", "corporateid": "CORP-821d5655a7dcd7f1017eaae1db60ffe6", "type": "BRANCH", "branch_name": "Факультет Інформатик", "branch_adress": "м.Київ", "branch_phone": "(050)-222-33-44", "branch_sign_certs": ["cert-ffffff", "cert-dddddddddddd", "cert-TTTTTTTTTTTTTT"], "branch_encrypt_certs": ["cert-vvvvvvvvvvvvvv"]}
```

```
INFO:test_getBranchItem:Перевіря відповідь statusCode=200
```

.

```
INFO:test_deleteBranchItem:Опис тесту: ===== Видалити один бренд в корпорації on id =====
```

```
INFO:test_deleteBranchItem:Формую запит на url: http://localhost:8080/branch-api/branch/CORP-821d5655a7dcd7f1017eaae1db60ffe6/BRANCH-3c94f4eaa5ece90467159915a0ba081f
```

```
INFO:test_deleteBranchItem:Method: http-get
```

```
INFO:test_deleteBranchItem:Виконю http запит
```

```
INFO:test_deleteBranchItem:Response: {"id": "BRANCH-
```

```
3c94f4eaa5ece90467159915a0ba081f", "rev": "2-affddde464b17463ce4e83702977f201", "ok": true}
```

```
INFO:test_deleteBranchItem:Перевіря відповідь statusCode=200
```

.

```
-----
Ran 4 tests in 0.967s
```

OK

