

process-api короткий опис та приклади unit тестів

Цей API розроблено для **банку**, щоб обробити платіжну відомість

Методи API

baseURL: /process-api/process

http-post - Встанвити статус обробки платіжної відомості з paymentid

Моделює обробку платіжної відомості на зарахування зарплати в банку

► Детально про метод

- обов'язкові http заголовки

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

- параметри запиту

В **path** передається ідентифікатор відомості paymentid

- запит

```
{
  "proc_status": "CHECKING",
  "proc_status_dsc": "Перевірка відомості",
  "proc_error_dsc": ""
}
```

- відповідь

```
{"paymentid": "PAYMENT-d6d4d6b42e93963c596fff4a65caf148"}
```

Unit-тести

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

- test_processPayment - встановити статус обробки платіжної відомості

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 TestProcessAPI(unittest.TestCase):
    baseurl="http://localhost:8080"
    requests={}
    requests["paymentid"]="PAYMENT-d6d4d6b42e93963c596fff4a65caf148"
    requests["processPayment"]={
        "proc_status": "CHECKING",
        "proc_status_dsc": "Перевірка відомості",
        "proc_error_dsc": ""
    }

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

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

    def test_processPayment(self):
        """===== Обробити відоміст на зарахування ====="""
        log=logging.getLogger('test_processPayment')
        log.info( f"Опис тесту: {self.shortDescription()}" )

        log.info( f"Отримую тестовий запит")
        req_obj = TestProcessAPI.requests["processPayment"]
        req_url=TestProcessAPI.baseurl + f"/process-
api/process/{TestProcessAPI.requests['paymentid']}"
        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.post(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 suite():
    suite = unittest.TestSuite()
    suite.addTest(TestProcessAPI('test_processPayment'))

    return suite

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

```
INFO:test_processPayment:Опис тесту: ===== Обробити відоміст на
зарахування =====
INFO:test_processPayment:Отримую тестовий запит
INFO:test_processPayment:Формую запит на url: http://localhost:8080/process-
api/process/PAYMENT-d6d4d6b42e93963c596fff4a65caf148
INFO:test_processPayment:Method: http-PUT
INFO:test_processPayment:Тіло запиту: {"proc_status": "CHECKING",
"proc_status_dsc": "Перевірка відомості", "proc_error_dsc": ""}
INFO:test_processPayment:Виконю http запит
INFO:test_processPayment:Response: {"result":"PAYMENT-
d6d4d6b42e93963c596fff4a65caf148"}
INFO:test_processPayment:Перевіря відповідь statusCode=200

.
-----
Ran 1 test in 1.032s

OK
```