



Сказ о том, как мы платежный шлюз тестируем

Сергей Озеранский

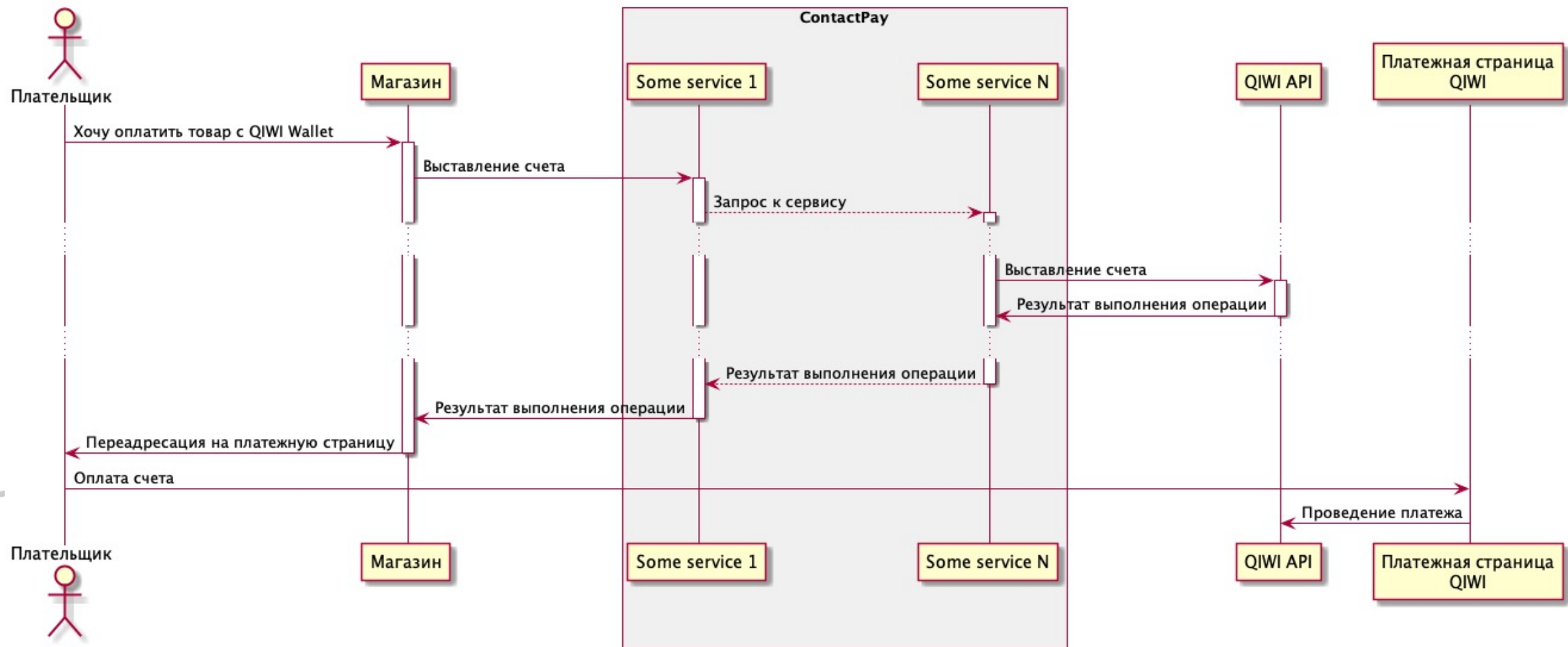


- 01 Самостоятельный FinTech-стартап внутри QIWI
- 02 Высокопроизводительный и отказоустойчивый платежный шлюз
- 03 Соответствуем стандартам безопасности PCI DSS
- 04 Микросервисная архитектура. Сервисы на Python 🐍
- 05 Много интеграций с внешними API

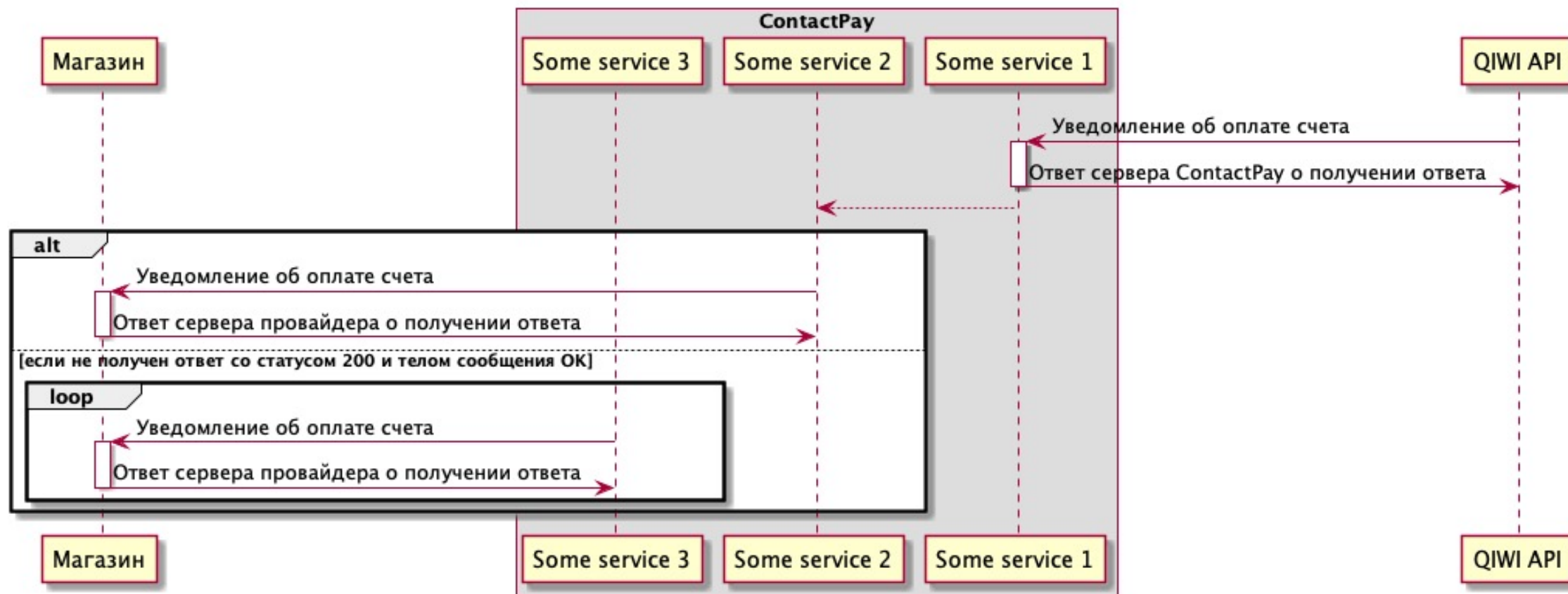
Метрики кода

- 01 Корректность (без багов)
- 02 Читаемость
- 03 Поддерживаемость
- 04 Быстродействие
- 05 Безопасность\Защищенность
- 06 Тестируемость

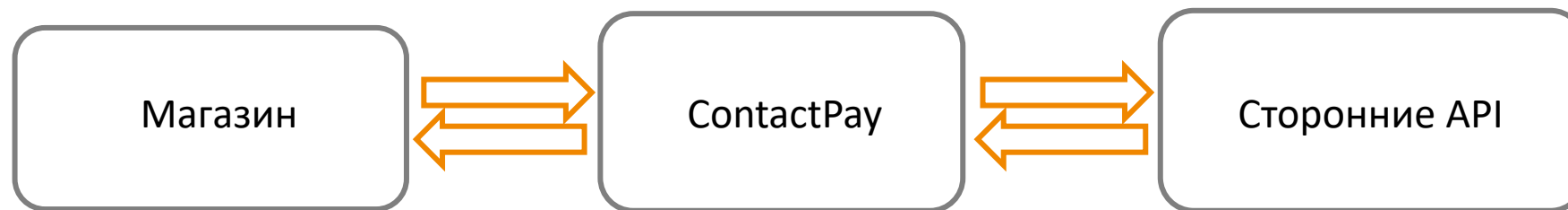
Выставление счета



Уведомление об оплате



По простому



Что хотим от тестирования

- 01 Максимально реалистично
- 02 Edge cases (ConnectionError, ...)
- 03 Проверять callback'и (наши и платежной системы)

Сложности и ограничения

- У стороннего сервиса нет песочницы
- Ограничения в песочнице
- Песочница может быть не доступна
- Не можем тестировать все Edge cases
- Не можем тестировать различные сценарии отправки callback'ов



Произвольный response

```
s.ozerskiy ~  
[ http '127.0.0.1:8001/common/custom-handler?status_code=201&content={"same_key":"same_value"}&content_type=application/json'  
HTTP/1.1 201 Created  
Connection: close  
Content-Type: application/json  
Date: Sun, 28 Nov 2021 21:47:44 GMT  
Server: WSGIServer/0.2 CPython/3.7.12  
Server-info: mock-service/common-custom-handler  
  
{  
  "same_key": "same_value"  
}
```

Request Tracker List

OPTIONS

GET



«

1

2

»

GET /api/v1/trackers/

HTTP 200 OK

Allow: GET, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
{
  "count": 14,
  "next": "http://127.0.0.1:8002/api/v1/trackers/?page=2",
  "previous": null,
  "results": [
    {
      "id": 14,
      "uid": "6aa74482-41f6-4df2-aec2-dbbe533640b0",
      "handler": "common-custom-handler",
      "request_info": {
        "body": "",
        "path": "common/custom-handler",
        "method": "GET",
        "headers": {
          "Host": "127.0.0.1:8001",
          "Accept": "*/*",
          "Connection": "keep-alive",
          "User-Agent": "HTTPie/2.6.0",
          "Content-Type": "text/plain",
          "Content-Length": "",
          "Accept-Encoding": "gzip, deflate"
        },
        "query_params": {
          "content": [
            "{\"same_key\":\"same_value\"}"
          ]
        }
      }
    }
  ]
}
```

- Webhook Sessions
- Request handlers

Mockingbird - easy mocking any requests

Meta information

Handler path
http://127.0.0.1:8001/hook_sessions/88b3dc05-1d5e-43e5-9003-b1f6103

Created at
2021-12-08T23:55:16.937350+03:00

Modified at
2021-12-09T00:02:38.037425+03:00

Name
exact aqua chinchilla

Response settings

Delay
0s

Status code
200

Body format
python

Body

```
1 body = []
2 fake = faker.Factory.create('ru_RU')
3
4 for i in range(1, 6):
5     body.append(
6         {'id': i,
7          'name': fake.name(),
8          'bithday': fake.date()}
9     )
10
11 response_info.content = body
```

Headers

```
1 {}
```

Scripting (action after response)

Delay

Code

```
1
```



```
1 // 20211209000241
2 // http://127.0.0.1:8001/hook_sessions/88b3dc05-1d5e-43e5-9003-b1f610384105
3
4 ▼ [
5 ▼ {
6     "id": 1,
7     "name": "Евсеев Ипполит Измаилович",
8     "bithday": "1991-03-28"
9 },
10 ▼ {
11     "id": 2,
12     "name": "Воронцова Акулина Аскольдовна",
13     "bithday": "2002-04-20"
14 },
15 ▼ {
16     "id": 3,
17     "name": "Ярополк Якубович Аксенов",
18     "bithday": "1980-05-17"
19 },
20 ▼ {
21     "id": 4,
22     "name": "Феофан Изотович Орлов",
23     "bithday": "1994-09-22"
24 },
25 ▼ {
26     "id": 5,
27     "name": "Галкина Раиса Юльевна",
28     "bithday": "1982-03-17"
29 }
30 ]
```



Meta information

Handler path

http://127.0.0.1:8001/handlers/invoice/[a-zA-Z0-9]

Created at

2021-12-09T00:11:24.247150+03:00

Modified at

2021-12-09T00:16:30.144762+03:00

Deactivated at

Name

simulate a third-party api

Response settings

Type path matching

regex

Path

/invoice/[a-zA-Z0-9]

Method

GET

Delay

0s

Status code

200

Body format

python

Body

```
1 fake = faker.Factory.create('ru_RU')
2
3 body = {
4     'id': request_info.path.split('/')[-1],
5     'description': fake.text(max_nb_chars=25),
6     'amount': fake.pyfloat(positive=True)
7 }
8
9 response_info.content = body
```

Headers

```
1 { }
```

```
1 // 20211209001634
2 // http://127.0.0.1:8001/handlers/invoice/qwerty123
3
4 {
5   "id": "qwerty123",
6   "description": "Сравнение товар крутой.",
7   "amount": 181254201.59564
8 }
```

Mockingbird - easy mocking any requests

Detailed information about the request tracker: c952efa5-6f20-4330-bf87-0212634ebb06

```
1 {
2   "id": 201,
3   "uid": "c952efa5-6f20-4330-bf87-0212634ebb06",
4   "handler": "",
5   "request_info": {
6     "body": "",
7     "path": "/invoice/qwerty123",
8     "method": "GET",
9     "headers": {
10      "Host": "127.0.0.1:8001",
11      "Accept":
12      "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9",
13      "Cookie": "csrftoken=vFmp3WAOdNZ2EGBZWYKP9JFunl9mMNH129w8JCRGA3Y35QNSv0XF8eOUHY2zxQ;
14      session=eyJjc3JmX3Rva2VuIjoimTJhOGM4OTg4NTE5NjQ1MTZlM2MxYzE0NTI0OWIwZWRkN2U1ZTQ5MCMJ9.YZv9kg.sZh2SCCs1O5F_NI0lsapNAsc
15      314"
```

```
class CommonData(BaseJsonPayload):
    same_key: pydantic.StrictInt

class CommonHandler(HTTPRequestHandler):
    """Custom handler.

    Examples:
        /common/custom-handler?status_code=201&content={"a":1}&content_type=application/json

    """

    name = 'common-custom-handler'
    matching_rules = [HTTPPathMatchRule(path='common/custom-handler')]
    validators = [RequestSchemaValidator(CommonData)]

    def create_error_response_info(
        self,
        request_info: RequestInfo,
        exception: ValueErrorType,
    ) -> ResponseInfo:
        return ResponseInfo(
            status_code=HTTPStatus.BAD_REQUEST,
            headers={'Content-Type': 'application/json'},
            content={
                'errors': get_pydantic_errors_description(exception),
            },
        )
```


Ошибка валидации

```
s.ozerskiy ~  
http '127.0.0.1:8001/common/custom-handler' same_key=same_value -v  
POST /common/custom-handler HTTP/1.1  
Accept: application/json, */*;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Content-Length: 26  
Content-Type: application/json  
Host: 127.0.0.1:8001  
User-Agent: HTTPie/2.6.0  
  
{  
  "same_key": "same_value"  
}  
  
HTTP/1.1 400 Bad Request  
Connection: close  
Content-Type: application/json  
Date: Sun, 28 Nov 2021 22:40:00 GMT  
Server: WSGIServer/0.2 CPython/3.7.12  
  
{  
  "errors": "same_key=\"value is not a valid integer\""  
}
```

Успешная валидации

```
s.ozerskiy ~  
└─ http '127.0.0.1:8001/common/custom-handler' same_key:=123 -v  
POST /common/custom-handler HTTP/1.1  
Accept: application/json, */*;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Content-Length: 17  
Content-Type: application/json  
Host: 127.0.0.1:8001  
User-Agent: HTTPie/2.6.0  
  
{  
  "same_key": 123  
}  
  
HTTP/1.1 200 OK  
Connection: close  
Content-Type: text/plain  
Date: Sun, 28 Nov 2021 22:39:37 GMT  
Server: WSGIServer/0.2 CPython/3.7.12  
Server-info: mock-service/common-custom-handler
```

```
s.ozerskiy ~  
http '127.0.0.1:8001/errors/internal_server'  
HTTP/1.1 500 Internal Server Error  
Connection: close  
Content-Type: text/html; charset=utf-8  
Date: Sun, 28 Nov 2021 22:47:14 GMT  
Server: WSGIServer/0.2 CPython/3.7.12  
server-info: mock-service/internal_server_error  
  
500 Internal Server Error
```

```
s.ozerskiy ~  
date && http '127.0.0.1:8001/common/slow-handler/20?status_code=201&content={"a":1}&content_type=application/json'  
понедельник, 29 ноября 2021 г. 01:44:59 (MSK)  
HTTP/1.1 201 Created  
Connection: close  
Content-Type: application/json  
Date: Sun, 28 Nov 2021 22:45:20 GMT  
Server: WSGIServer/0.2 CPython/3.7.12  
Server-info: mock-service/common-slow-handler  
  
{  
  "a": 1  
}
```

```
class DelayCallback(DelayRequestAction):
    delay_time = 5
    url = 'http://127.0.0.1:8001/common/custom-handler?content=callback'

    def get_request_params(
        self,
        request_info: RequestInfo,
        response_info: ResponseInfo,
    ):
        return {
            'data': {'hello': 'callback'},
        }
```

```
class CommonHandler(HTTPRequestHandler):
    """Custom handler.

    Examples:
        /common/custom-handler?status_code=201&content={"a":1}&content_type=application/json

    """

    name = 'common-custom-handler'
    matching_rules = [HTTPPathMatchRule(path='common/custom-handler')]
    actions = [DelayCallback()]
```

Больше, чем хотели

01 

Автотесты

Покрываем больше тест кейсов, если бы mock сервиса не было

02 

Нагрузочное тестирование

Можем проводить нагрузочное тестирование без ограничений и зависимостей

03 

Sandbox API ContactPay

Появился независимый от сторонних сервисов Sandbox



Сергей Озеранский

