LavaBusiness

Iztrox

дек. 13, 2022

Оглавление

Асинхронный клиент для работы с Lava.ru Бизнес-API

Глава 1

Requirements

```
httpx>=0.23.1
```

Глава 2

Установка

Через pip:

```
pip install lavabusiness
```

Через Git:

```
git clone https://github.com/lztrox/lavabusiness
cd lavabusiness
python setup.py install
```

2.1 Асинхронный клиент

2.2 Types

2.3 Примеры

Создание счета

```
import asyncio
from LavaBusiness import AioLava

SECRET_KEY = ""
PROJECT_ID = ""

api = AioLava(SECRET_KEY, PROJECT_ID)

```

(continues on next page)

(продолжение с предыдущей страницы)

```
async def main():
    invoice = await api.create_invoice(100)
    print(f'Pay url: {invoice.url}')
    print(f'Invoice_id: {invoice.invoice_id}')

asyncio.run(main())
```

Проверка статуса счета

```
import asyncio
from LavaBusiness import AioLava

SECRET_KEY = ""
PROJECT_ID = ""

api = AioLava(SECRET_KEY, PROJECT_ID)

async def main():
    INVOICE_ID = ""
    status = await api.invoice_status(INVOICE_ID)

    if status == 'success':
        print('Счет оплачен')
    elif status == 'expired':
        print('Счет просрочен')
    else:
        print('Счет ожидает оплаты')

asyncio.run(main())
```

Получение доступных методов оплаты

```
import asyncio
from LavaBusiness import AioLava

SECRET_KEY = ""
PROJECT_ID = ""

api = AioLava(SECRET_KEY, PROJECT_ID)

async def main():
    pay_methods = await api.pay_methods()
    for method in pay_methods:
        print(method)

asyncio.run(main())
```

Получение баланса магазина

```
import asyncio
from LavaBusiness import AioLava
```

(continues on next page)

(продолжение с предыдущей страницы)

```
SECRET_KEY = ""
PROJECT_ID = ""

api = AioLava(SECRET_KEY, PROJECT_ID)

async def main():
    balances = await api.balance()
    print(balances['balance'])
    print(balances['freeze_balance'])

asyncio.run(main())
```

Создание вывода

```
import asyncio
from LavaBusiness import AioLava

SECRET_KEY = ""
PROJECT_ID = ""

api = AioLava(SECRET_KEY, PROJECT_ID)

async def main():
    payoff_id = await api.create_payoff(100)
    print(f'Payoff ID: {payoff_id}')

asyncio.run(main())
```

Проверка статуса вывода

```
import asyncio
from LavaBusiness import AioLava

SECRET_KEY = ""
PROJECT_ID = ""

api = AioLava(SECRET_KEY, PROJECT_ID)

async def main():
    PAYOFF_ID = ""
    status = await api.payoff_status(PAYOFF_ID)

    if status == 'success':
        print('Вывод успешно завершен')
    elif status == 'rejected':
        print('Вывод отменен')
    else:
        print('Вывод в очереди')

asyncio.run(main())
```

Получение тарифов на вывод

```python
import asyncio
from LavaBusiness import AioLava

SECRET_KEY = ""
PROJECT_ID = ""

api = AioLava(SECRET_KEY, PROJECT_ID)

async def main():
    payoff_tarrifs = await api.payoff_tarrifs()
    for tarrif in payoff_tarrifs:
        print(tarrif)

asyncio.run(main())
```