

## [OCTO-943] [Retail] Раздел PAY Paynet оплата коммуналки и т.д. Created: 29/Sep/25 Updated: 16/Oct/25

Status:	Analysis
Project:	Octo
Components:	None
Affects versions:	None
Fix versions:	None
Security Level:	OCTO

Type:	Epic	Priority:	Highest
Reporter:	khusan.ya	Assignee:	Ilya Zinovev
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Attachments:	UZPaynet API JSON Gateway v3 (239681fe-b16d-44d0-8bf7-f4a1ced4398f).pdf  image-20250930-113721.png image-20250930-105335.png  JSON GW2.postman_collection (d101d582-4e39-4556-822d-b7467bddedf3).json  268c2797-1a07-4edc-aa14-55b47c3c438c.mp4
Sprint:	
Client:	OCTO

## Description

- 1. История изменений
- 2. Описание
- 3. Системы/термины
- 4. User Stories
  - Story 1: Оплата услуги
  - Story 2: История платежей
  - Story 3: Настройка комиссий платежей
  - Story 4: История платежей бэк-офис
- 5. API-флоу
- 6. НФТ
- 7. Ограничения/допущения
- 8. Приложение

## 1. История изменений

Версия	Дата изменения	Описание изменений
1.0	2025-09-30	Initial

## 2. Описание

Прием платежей от пользователей кошелка за услуги через шлюз PAYNET.

## 3. Системы/термины

Кошелек Астериум (веб, мобилки)

Админка кошелька Астериум

Paynet API - платежный шлюз для оплат услуг (сотовая связь, коммуналка и т.п.):

- Category - категория поставщиков
- Provider - поставщик услуг оплат
- Services - сервисы поставщика. Info - для получения информации по лицевому счету, Payment - осуществление платежа

Агент - пользователь шлюза, Астериум

## 4. User Stories

## Story 1: Оплата услуги

Как пользователь, хочу оплатить услугу через приложение, чтобы получить доступ к сервису.

### Основной сценарий :

- Выбор сервиса (ручной, поиск)
- Ввод реквизитов (лицевой счёт, договор и т.п.)
- Просматривает детали лицевого счета
- Ввод суммы;
- Выбор токена
- Подтверждение оплаты
- Получение статуса платежа
- Просмотр деталей чека (терминалы, фискализация и т.п.)
- При успешной оплате транзакция сохраняется в историю
- Пользователь сохраняет шаблон платежа (опционально)

### Альтернативный сценарий (оплата по шаблону)

- Выбирает шаблон
- Ознакомляется с деталями платежа
- Производит оплату

### AC

#### 1. Выбор сервиса

- Пользователь может выбрать сервис из списка категорий.
- Пользователь может найти сервис через поиск.
- В избранных отображаются ТОП-10 платежей пользователя по общему количеству транзакций

#### 2. Ввод реквизитов

- Кнопка “Детали” отображается только для тех провайдеров, у которых есть info-сервис
- После ввода реквизитов на событии on-out из поля, отправляется запрос на получение данных лицевого счета. Если запрос успешен и пришли детали, кнопка “Детали” становится активной. Если запрос отработан с ошибкой, то показать поп-ап с ошибкой, например “Логин некорректен”
- При некорректных реквизитах кнопка остаётся неактивной, а пользователю отображается ошибка.
- Реализована валидация полей согласно REGEXP из ответа на получение списка полей поставщика
- По обращению к кнопке “Детали” пользователю отображается информация по счёту (баланс и другие доступные детали из ответа)\*

\*детали - опциональный функционал. Для некоторых сервисов PAYNET получение деталей по лицевому счету недоступно.

#### 1. Оплата

- После подтверждения выполняется конвертация по специализированному курсу обмена для платежей
- Пользователь видит статус платежа (успешно / ошибка)
- При успешной оплате формируется чек (с деталями терминала, фискализацией и пр.) и сохраняется в историю транзакций

#### 2. Шаблоны

- Пользователь может сохранить реквизиты как шаблон.
- Шаблон содержит все необходимые данные для повторного платежа.
- После выбора шаблона пользователю отображаются детали шаблона.
- Далее как в стандартном флоу оплаты
- Пользователь может удалить шаблон

## Story 2: История платежей

Как пользователь, хочу просматривать историю платежей, чтобы анализировать расходы.

### Основной сценарий :

- Вход в раздел история
- Переход по вкладке “История платежей”
- Подгружается список транзакций

### AC

1. Пользователь может открыть раздел История → История платежей.
2. При отсутствии транзакций отображается экран-заглушка («У вас пока нет платежей»).
3. Система отображает список транзакций пользователя с базовыми атрибутами:
  - Айди транзакции
  - Категория (связь, холодная вода, игры и т.п.)

- Поставщик (beeline, steam, гор газ и т.п)
  - Токен оплаты
  - Сумма платежа
  - Дата транзакции
  - Статус платежа
4. Пользователь может кликнуть на транзакцию и увидеть детали:
- Атрибуты с пункта 2
  - Реквизиты платежа
  - Фискальные данные (ID, фискального чека, статус фискализации)
5. Доступна фильтрация по категории, поставщику, дате, статусу

### Story 3: Настройка комиссий платежей

Как сотрудник бэк-офиса, хочу настраивать комиссии для обмена в рамках платежей, чтобы кастомизировать прибыль для платежей.

#### Основной сценарий

1. Сотрудник авторизуется в админке.
2. Переходит в раздел Платежи → Настройка платежей → Комиссии
3. Выбирает платежный шлюз (paynet, octogames)
4. Устанавливает процент комиссии для каждого шлюза
5. Сохраняет настройки

#### AC

- Отдельный скрипт “Настройка платежей” доступен в админке. Далее данный скрипт можно прокрутить к определенной роли
- После сохранения настроек комиссии, они применяются для новых платежей

### Story 4: История платежей бэк-офис

Как сотрудник бэк-офиса, хочу просматривать историю платежей пользователей, чтобы делать сверки с партнерами

#### Основной сценарий

1. Сотрудник авторизуется в админке.
2. Переходит в раздел Платежи → История платежей
3. Выгружает отчеты

#### AC

- Запись содержит атрибуты:
  - Пользователь
  - Айди транзакции
  - Категория (связь, холодная вода, игры и т.п.)
  - Поставщик (beeline, steam, гор газ и т.п.)
  - Платежный шлюз (paynet, octogames)
  - Сумма платежа в UZS
  - Дата транзакции
  - Статус платежа
- Есть возможность выгрузки отчета в форматы excel, csv
- Реализована фильтрация по дате, платежному шлюзу, статусу
- Отдельный скрипт “История платежей” доступен в админке. Далее данный скрипт можно прокрутить к определенной роли

## 5. API-флоу



Рис. 1. Схема взаимодействия без кэширования данных из платежного шлюза

#### Дополнение к схеме

По структуре, есть список категорий поставщиков. Из каждой категории по айди категории можно получить список поставщиков. По айди поставщика получаем доступные сервисы поставщика. Сервисы которые нам нужны:

- info-сервис для получения полей, необходимых для запроса деталей лицевого счета. Находим по атрибуту title

```

    "providerId": 4135,
    "id": 6766,
    "titleRu": "Проверка счёта Uzonline",
    "titleEn": "Проверка счёта Uzonline",
    "titleUz": "Raqamni tekshirish Uzonline",
    "childId": 6767,
    "order": 834,
    "mcc": null
},
{
    "providerId": 4135,
    "id": 6767,
    "titleRu": "Uzonline (интернет)",
    "titleEn": "Uzonline (internet)",
    "titleUz": "Uzonline (internet)",
    "childId": 0,
    "order": 835,
    "mcc": "4814"
}
]
}
{
    "jsonrpc": "2.0",
    "id": 123,
    "result": {
        "services": [
            {
                "providerId": 1465,
                "id": 2273,
                "titleRu": "Получение информации",
                "titleEn": "Получение информации",
                "titleUz": "Maъlumot olish",
                "childId": 2173,
                "order": 347,
                "mcc": null
            },
            {
                "providerId": 1465,
                "id": 2173,
                "titleRu": "Оплата",
                "titleEn": "Onlata",
                "titleUz": "To'lov",
                "childId": 0,
                "order": 342,
                "mcc": "4899"
            }
        ]
    }
}

```

- pay-сервис для получения полей, необходимых для самой оплаты. Находим его по mcc != null .

По айди необходимых сервисов, получаем данные из getServiceInfo:

- Лимиты
- Необходимые поля. Отображаем только поля isRequired = true
- Валидации и тип полей (подробнее в приложении)

Для info и pay запросов вызывается метод performTransaction:

- для info передается айди info-сервиса
- для pay передается айди pay-сервиса

## 6. НФТ

- Установит rate-limit на все запросы: 1 запрос в секунду на каждого пользователя.

## 7. Ограничения/допущения

После получения тестовых доступов, возможны некоторые поправки и дополнения описания атрибутов.

## 8. Приложение

### Документация PAYNET API

[UZPaynet API JSON Gateway v3 \(239681fe-b16d-44d0-8bf7-f4a1ced4398f\).pdf](#)

### Коллекция запросов PAYNET API

[JSON GW2.postman\\_collection \(d101d582-4e39-4556-822d-b7467bddedf3\).json](#)

### Тестовые данные

base\_url: <http://gw2-test.paynet.uz:8588>

"username": "username\_17627\_56530",  
 "password": "password\_17627\_56530",  
 "terminal\_id": 4027096

### Дизайн

<https://www.figma.com/design/EdVpLx1Bw2S6PH9YLSBVUv/Web-app-%7C-Asterium-updates?node-id=16789-356717&p=f&t=xk4by8F1SB4E68Pu-0>

Ссылка на исходную документацию

<https://asteriumuz.atlassian.net/wiki/spaces/~712020e44555f5d9e24f7ea8cd92b801c8f85b/pages/111804468/Paynet#af612161-2256-456b-b8e9-639b0e4ae6a4>

Кикофф

## Comments

Comment by [khusan.ya](#) [ 29/Sep/25 ]

[Pavel Gusarov Alexey Venevtsev](#) нужно брать в анализ в этом спринте

Comment by [khusan.ya](#) [ 30/Sep/25 ]

Эпик разделим на 2 части оплата игр через окто будет в другом эпике

Comment by [Pavel Gusarov](#) [ 15/Oct/25 ]

Кикоф прошел 15.10 в 13:30. Вернули на анализ. По результатам доработки ТЗ готово для декомпозиции и оценки

Comment by [Pavel Gusarov](#) [ 15/Oct/25 ]

[khusan.ya Denis Kulchak Ilya Zinovev](#)

Коллеги, на сколько вижу, по дизайну появились комментарии в фигме. Стоит ли их учесть перед тем, как отдавать задачу в разработку?

Предлагаю, чтобы [Ilya Zinovev](#) поставил задачу на дизайнера в рамках этого эпика, если это необходимо.

Comment by [Denis Kulchak](#) [ 16/Oct/25 ]

[Pavel Gusarov](#), да комменты появились, но к ним есть вопросы, как только определимся сразу задачу заведм, спасибо

Generated at Fri Oct 17 15:27:17 UTC 2025 by Alexey Venevtsev using Jira 1001.0.0-SNAPSHOT#100290-

rev:7711952900727759028b2799f351c92c8b8a2417.