

# Цифровая маркетинговая платформа ПэйТую

Функциональные возможности

Автор: ООО «Пэйтую»

Дата создания: 21.03.2023

# Содержание

<b>1 Термины и сокращения</b>	<b>3</b>
<b>2 Дополнительные документы</b>	<b>4</b>
Свидетельство о государственной регистрации программы для ЭВМ	4
<b>3 Введение</b>	<b>5</b>
<b>4 Функциональные возможности</b>	<b>6</b>
4.1 Цели и назначение	6
4.2 Возможности участников	7
4.2.1. Клиент	7
4.2.2. Оператор платформы	8
4.2.3. Паблишер	8
4.2.4. Партнер	9
4.3 Поддерживаемые клиентские механики	9
4.4 Развитие партнерской сети и подключение Партнеров	10
<b>5 Внедрение Витрины предложений</b>	<b>11</b>
5.1 Авторизационный токен JWT	11
Декодированное тело токена	11
Закодированный токен	11
Пример создания на Node.js	11
5.2 Автофокус категории по Alias	12
Пример реализации Alias	12
5.3 Обработка стрелки “Назад” и закрытие Витрины	12
Пример реализации на iOS	13
Пример реализации на Android	13
Необходимые настройки webview	13
Запуск webview с ссылкой на витрину и закладывание интерфейса для связи JS кода с нативом Android	14
5.4 Открытие ссылки во внешнем браузере	14
Пример реализации перехвата ссылки	14

# 1 Термины и сокращения

Термин	Описание
Оператор платформы	Владелец системы, агрегирующий договоры и технологические стыки с другими участниками
Публишер	SaaS клиент, компания, использующая решение PAY2U для реализации собственных проектов лояльности в рамках своего основного Продукта
Партнер	Компания, размещающая свои предложения (оферы) на Платформе с целью предоставления их аудитории Публишеров

## 2 Дополнительные документы

Свидетельство о государственной регистрации программы для ЭВМ

**Название** - "Цифровая маркетинговая платформа PAY2U"

**Номер свидетельства** - 2017613139

**Номер заявки** - 2016661312

**Дата поступления** - 25 октября 2016

**Дата государственной регистрации в Реестре программ для ЭВМ** - 13 марта 2017

**Правообладатель** - "Общество с Ограниченной Ответственностью ПэйТую"

### 3 Введение

Данный документ описывает функциональные возможности, цели и назначения программного обеспечения «Цифровая маркетинговая платформа Пэйтую», разработанного ООО «Пэйтую».

## 4 Функциональные возможности

### 4.1 Цели и назначение

Платформа PAY2U является SAAS системой с многопользовательским доступом, обеспечивающим построение Whitelabel программ лояльности и партнерских программ (ПП) с возможностью глубокой кастомизации ПП под потребности Whitelabel партнера.

Платформа позволяет агрегировать в одном месте предложения различных Партнеров и Агрегаторов, и предоставлять к ним доступ в рамках Whitelabel ПЛ. Предложения Партнеров могут иметь различные мотивационные механики, такие как скидка, либо кешбэк за те и или иные действия клиентов.

На верхнем уровне определены следующие роли участников:

- Оператор платформы;
- Пабlishер;
- Партнер.

Отношения между участниками определены в соответствующих договорах. Обобщенная схема взаимодействия при начислении кешбэка приведена ниже:

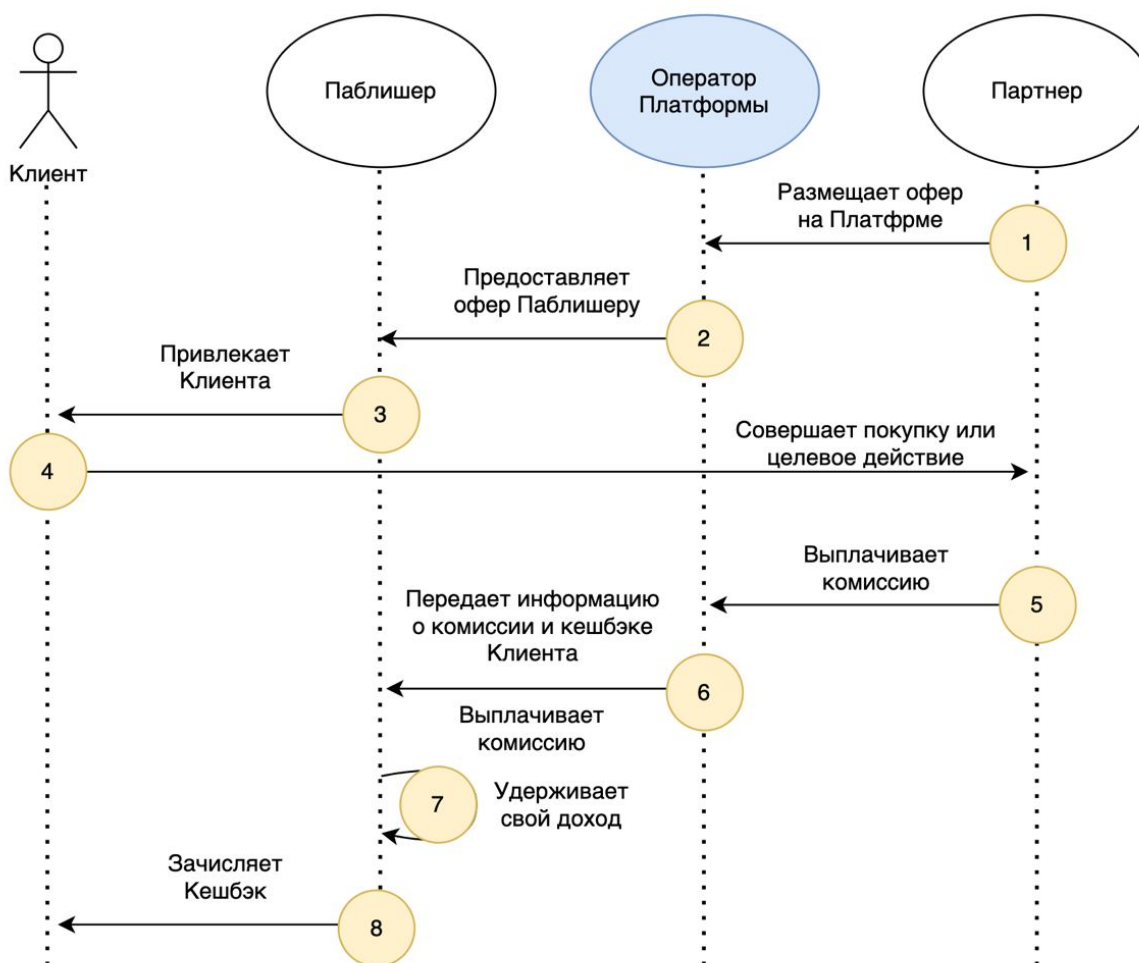


Рисунок 1 - Схема взаимодействия между участниками

## 4.2 Возможности участников

### 4.2.1. Клиент

Клиент взаимодействует с платформой при помощи Витрины предложений, которая реализует всю функциональность клиентского Продукта и предоставляет клиенту доступ к следующей функциональности:

- Просмотр категорий предложений
- Просмотр предложений внутри выбранной категории
- Поиск предложений по набранному тексту в строке поиска
- Поиск предложений по выбранным фильтрам
- Просмотр условий получения вознаграждения за использование предложения
- Переход на сайт Партнера при использовании предложения
- Выпуск купона/промокода для использования на сайте Партнера при использовании предложения

- Бронирование столика в ресторане при использовании предложения

## 4.2.2. Оператор платформы

Предоставлены следующие инструменты для ведения бизнеса по модели SaaS (Software as a Service):

- Инструменты развертывания окружения для SaaS клиента (Пабlishера);
- Развертывание клиентского фронтенда, соответствующего Пабlishера и генерирование параметров авторизации;
- Стилизация Витрины Предложений Пабlishера и конфигурирование внешнего вида с минимальным привлечением команды разработки;
- Создание учетных записей для Пабlishера для доступа к админ-панели;
- Подключение интеграционных стыков, позволяющие выполнять начисления средств на балансы пользователей Пабlishера;
- Предоставление Пабlishеру доступа к аналитическим данным, а также настройка выгрузки статистики в собственные BI системы Пабlishера;
- Инструменты централизованного управление доступностью оферов для Пабlishеров;
- Процессинг кешбэк транзакций, обеспечивающий взаиморасчеты как с Пабlishерами, так и с Партнерами;
- Инструменты обработки клиентских обращений, поступающих с первой линии поддержки Пабlishера;
- Инструменты развития партнерской сети, позволяющие интегрировать Партнеров на Платформу;
- Инструменты автоматизации сверок и взаиморасчетов.

## 4.2.3. Пабlishер

Платформа предоставляет возможность создавать разделы лояльности с партнерскими предложениями с кешбэком и скидками.

Пабlishеру доступна Витрина, реализующий все пользовательские сценарии, интегрируемый в мобильное приложение или на сайт, и не требующий дополнительной разработки командой Пабlishера.

Пабlishер имеет полный контроль над представлением раздела и может конфигурировать весь контент Витрины:

- Настраивать категории и их внешний вид на главном экране раздела;



- Включать/Выключать оферы в соответствующие категории;
- Управлять контентом оферов;
- Создавать дополнительные блоки для продвижения (например раздел Популярное).

Кешбэк начисляется в автоматическом режиме либо на счет клиента, либо на счет бонусного кошелька. Платформа обеспечивает полную автоматизацию и прозрачность начисления в разрезе каждой транзакции.

Публишер имеет возможность управлять ставкой кешбэка, и тем самым зарабатывать с каждой транзакции, оставляя часть кешбэка в качестве дохода.

Помимо этого Публишеру доступны отчеты платформы и аналитическая информация, позволяющая развивать данный раздел и следить за ключевыми метриками Продукта.

Для запуска проекта Публишеру достаточно интегрировать web-view в свое мобильное приложение и настроить обмен данными с Платформой для зачисления кешбэка на счета/балансы/кошельки пользователей.

#### 4.2.4. Партнер

Для партнеров предоставляется возможность использовать Платформу в качестве инструмента привлечения пользователей из аудитории Публишера путем размещения оферов в супераппах Публишеров, проведения таргетированных кампаний, проведения совместных акций с Публишером.

### 4.3 Поддерживаемые клиентские механики

- Кешбэк при переходе на сайт/в приложение Партнера;
- Выпуск электронного промокода и получение скидки в он-лайн магазине/сервисе Партнера;
- Выпуск графического (штрих, QR) кода на скидку для предъявления на кассе офлайн-магазина Партнера;
- Скидка при переходе на специальный лендинг Партнера;
- Кешбэк при бронировании стола в ресторане через форму внутри Витрины;
- Кешбэк при оформлении заказа или покупку через миниап Партнера.

## 4.4 Развитие партнерской сети и подключение Партнеров

Технологическое подключение Партнера состоит в разработке интеграционных модулей в двух точках интеграции:

- Подключение к фиду оферов Партнера. Фид может быть представлен в любом формате: HTML, CSV, XML, JSON либо любой другой формат, предоставляемый Партнером;
- Подключение к статистике по совершенным заказам. Статистика может быть предоставлена в любом формате: HTML, CSV, XML, JSON либо любой другой формат, предоставляемый Партнером.

## 5 Внедрение Витрины предложений

Внедрение Витрины предложений производится Партнером в его мобильное приложение с помощью технологии WebView. Для идентификации клиента используется JWT-токен.

### 5.1 Авторизационный токен JWT

Декодированное тело токена

```
{
  "version": "1",
  "key": "USotRmVFsT4LvPlP6YffCtVuzMzYbPMm",
  "client_id": "27229974-8698-4342-b9e1-d7e48365cf77",
  "iat": 1626265847
}
```

Закодированный токен

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ2ZXJzaw9uIjoiaSI6ImtleSI6ImlVTb3R  
SbVZGc1Q0TH  
ZQbFA2WWZmQ3RwdXpNelliUE1tIiwiaWY2xpZW50X2lkIjoiaSI6ImtleSI6ImlVTb3R  
I5ZTEtZDdlND  
gzNjVjZjc3IiwiaWF0IjoxNjI5NDUyMjE2fQ.17XonhcmgFB2hg_WU7349x6LlCegLy-gQ6o  
umZmHGps
```

Пример создания на Node.js

```
const jwt = require('jsonwebtoken');
const PAY2U_SECRET = "KEEP_THIS_SECRET_SAFE";
const body = {
  version: "1",
  key: "USotRmVFsT4LvPlP6YffCtVuzMzYbPMm",
  client_id: "27229974-8698-4342-b9e1-d7e48365cf77",
}

const token = jwt.sign(body, PAY2U_SECRET, { algorithm: "HS256" });
```

В токене необходимо передать

- **version** - строка с версией интерфейса токена (в нашем случае это версия "1" )
- **key** - строка с ключом-идентификатором, полученным от PAY2U

- **client\_id** - строка-идентификатор клиента, который будет в реестре транзакций при сверке. Длина строки от 1 до 45 символов. Передавайте тот **client\_id**, по которому получится однозначно определить клиента

Также должно быть поле **iat** , содержащее **unix timestamp** времени выпуска токена в секундах. Сторонние библиотеки, как правило, заполняют это поле самостоятельно.

Токен подписывается алгоритмом **HS256** симметричным шифрованием. В интеграциях с сервисами **PAY2U** токен передается:

- либо в заголовке **HTTP** запроса ключом **X-Pay2u-Access-Token**
- либо в одном из **GET** параметров под ключом **access\_token**

## 5.2 Автофокус категории по Alias

Автофокус на выбранной категории происходит при добавлении к **url** сайта дополнительного **get** параметра **category\_focus** значение которого заранее устанавливается в Админ Консоли у необходимой категории.

Пример реализации **Alias**

[https://dev2.pay2u.ru/demo/partner\\_name/?focus\\_category=restaurants#/](https://dev2.pay2u.ru/demo/partner_name/?focus_category=restaurants#/)

## 5.3 Обработка стрелки “Назад” и закрытие Витрины

При необходимости закрытия **webview** Витрины и возврата в главное приложение необходимо внедрить в приложение скрипт для отлова событий по нажатию на стрелку **Назад**.

## Пример реализации на iOS

```

func injectCloseScript(_ webView: WKWebView) {
    let contentController = webView.configuration.userContentController
    contentController.removeScriptMessageHandler(forName: Constants.backHandlerName)
    if let view = presenter.view {
        contentController.add(view, name: Constants.backHandlerName)
        let script = """
            window.addEventListener('click', function(event) {
                window.webkit.messageHandlers.\(Constants.backHandlerName).postMessage(event.target.id);
            })
        """
        webView.evaluateJavaScript(script) { (_, error) in
            if let error = error {
                print(error)
            }
        }
    }
}

fileprivate struct Constants {
    private init() {}
    static let backHandlerName = "buttonClicked"
    static let externalItemName = "view_type"
    static let externalItemValue = "external"
}

```

## Пример реализации на Android

### Необходимые настройки webView

```

holder.webView.apply { this: WebView
    settings.javaScriptEnabled = true
    settings.setAppCacheEnabled(false)
    settings.domStorageEnabled = true
    settings.cacheMode = WebSettings.LOAD_NO_CACHE
    overScrollMode = View.OVER_SCROLL_NEVER

    holder.progressBar.design = RedCustomProgressBarDesign()

    viewModel.pay2YouViewStateLiveData.observe(viewLifecycleOwner, Observer { render(it) })
}

```

Запуск webview с ссылкой на витрину и закладывание интерфейса для связи JS кода с нативом Android

```

class JavaScriptInterface internal constructor(c: Context) {
    var mContext: Context

    /**
     * вызов функции с JS to native android при нажатии кнопки с id="escape-app-button"
     * назад на главный экран
     */
    @SuppressWarnings(...value: "unused")
    @JavascriptInterface
    fun clickEscapeAppButton() {
        viewModel.goBack()
    }

    init {
        mContext = c
    }
}

holder.webView.addJavascriptInterface(JavaScriptInterface(requireContext()), JS_BRIDGE_INTERFACE)
holder.webView.loadUrl(url, map)
}

```

## 5.4 Открытие ссылки во внешнем браузере

В Витрине присутствует возможность открытия предложений либо внутри webview, либо во внешнем браузере - это конфигурируется в бандле Витрины Предложений Пабличера. При выборе внешнего браузера ссылки будут иметь ключ `&view_type=external`, а при выборе открытия в webview ссылки будут иметь ключ `&view_type=internal`. Для корректного открытия ссылок приложение должно уметь их перехватывать.

Пример реализации перехвата ссылки

```

func navigationAction(for webView: WKWebView, action: WKNavigationAction, completion: ((Bool) -> Void)) {
    if action.navigationType == .linkActivated, let url = action.request.url {
        completion(false)
        let components = URLComponents(url: url, resolvingAgainstBaseURL: false)
        if let viewType = components?.queryItems?.first(where: { $0.name == Constants.externalItemName }),
            viewType.value == Constants.externalItemValue {
            UIApplication.shared.open(url)
        } else {
            webView.load(action.request)
        }
    }

    return
}

completion(true)
}

```