

# Pay One

## Payment Platform

v1.0a

October 2018

Protocol One

# Who Might Be Interested in This Document

Data presented in this document is relevant to a wide range of video game market participants:

- publishers;
- game developers;
- publishing platforms;
- technical experts;

This document describes the technical aspects of the [Pay One Payment platform](#) component of the Protocol One ecosystem. Description of the Ecosystem can be found within the [Protocol One Whitepaper](#) document.

## Introduction

Pay One is an open source service for accepting payments, which includes: a flexibly customizable payment form, a set of integration protocols, a management interface, and statistics tools.

Our mission is to give developers, publishers, platforms and stores an opportunity to accept payments with minimal losses and costs due to high conversion, low commission and minimization of labor costs related to integration and maintenance.

Our solution is a part of Protocol One IaaS, it is actively used in [Storefront constructor](#). Pay One can be installed as a component on a Protocol One-independent system, employing its own hardware or cloud platform.

Key features:

- **Speed and ease of integration.** Integrating payment acceptance through Pay One into your website is really easy. To install the widget on the page you only have to insert a single line of code. Integration tests are automated. KYC procedure and signing of a standard-form contract on payment acceptance occur within a few days. Electronic signature can be used.
- **Operation speed.** We believe that fast loading of the payment form is one of the most important factors for a good paying conversion. The approaches and technical solutions used in Pay One allow you to load a payment form very quickly, minimizing the wait on the side of the end user.
- **Accepting payments from users all over the world.** Pay One supports all payment methods necessary to accept payments from users anywhere in the world. Any combination of jurisdictions of the rights holder, card issuer and your company is possible (we cannot guarantee operation in areas where financial transactions are limited by international sanctions).

- **High high transaction completion percentage with minimal fraud.** We work with the most innovative suppliers of antifraud solutions. Our goal is, on the one hand, to maximize the difficulty of processing suspicious payments, and on the other, to minimize the amount of necessary confirmations when making a payment for a bona fide user.
- **Transparency of the legal system.** The money received from Pay One go with all the necessary documents for your bank.
- **VAT payment.** Pay One specialists are well versed in all the nuances of VAT payment in all relevant jurisdictions. We assume the role of an agent and give a comprehensive, understandable and transparent report on all tax payments.
- **Fast withdrawal.** A Pay One merchant may request a payout at any time without needing to wait till the end of an accounting period. The hold period between the arrival of funds and their activation for withdrawal is no more than 7 days.
- **Customization.** The visual style of the payment form can be customized at integration. The layout and the order of the fields and screens can be flexibly configured in the merchant's account for every project independently.
- **Localization and region-specific features.** The look of the payment form and the selection of final payment methods can be independently configured for each country or region. We support third-party developers and use best practices for different regions in Pay One.
- **Keeping up with local legislation.** We will help you to avoid breaking local laws by not allowing users to pay for products that are illegal in their region (for example, loot boxes in Belgium).
- **Open source code.** Our payment solution has an open source code. You can deploy a similar solution on your hardware. In this case, you will be free from commissions or any other payments to Pay One.
- **Low commissions.** We work with banks and other financial institutions directly. This direct cooperation allows us to reduce commissions. A large number of merchants in the system will help you get the best conditions thanks to monthly payment volumes. With this, the commission of the Pay One platform itself is minimal.
- **Detailed statistics and clear reports.** Pay One includes all the necessary tools for managing and monitoring user payments. In case you want to create more complex reports or use an external BI system for reports, we have Data API.
- **Transparent payout calculation principles.** Pay One reports contain complete, detailed payout information for each transaction.
- **Integration with other components of the Protocol One ecosystem.** As part of the Protocol One ecosystem, Pay One exchanges data with other components based on Open GDE. This allows you to take advantage of personalization mechanisms for fine-tuning the interface, a common storage of saved payment details for different merchants, regional prices and personal [Qilin](#) discounts.

## Roadmap

At this point Pay One is under active development, preparing for a public launch. The roadmap for the next iterations is outlined below.

Version	Date	Key tasks
1.0a	December 2018	Pay One specifications, onboarding, card payment acceptance, payment acceptance form with basic UI
1.0b	February 2019	Products and regional prices, more payment methods
1.0rc	May 2019	Dashboards, calculations and taxes
1.0r	August 2019	Commercial launch. Withdrawing funds, billing

Roadmap	3
Getting Started	6
Integration	6
Accepting a Payment	7
Projects and Products	8
Payment Form	9
Embed Options	12
Payment Methods	12
Localization	12
Retention of Users with Unfinished Transactions	13
Payment API	13
Payment Processing	14
Simple HTML Form	14
Calling JSON API	15
Payment Widget	15
Settlement of Disputes	16
Personal Account	17
Data API	19
Integration with Other Elements of Protocol One	19
Open Source Code	20
Disclaimer	20
Appendix 1. Payment Processing Charts	21

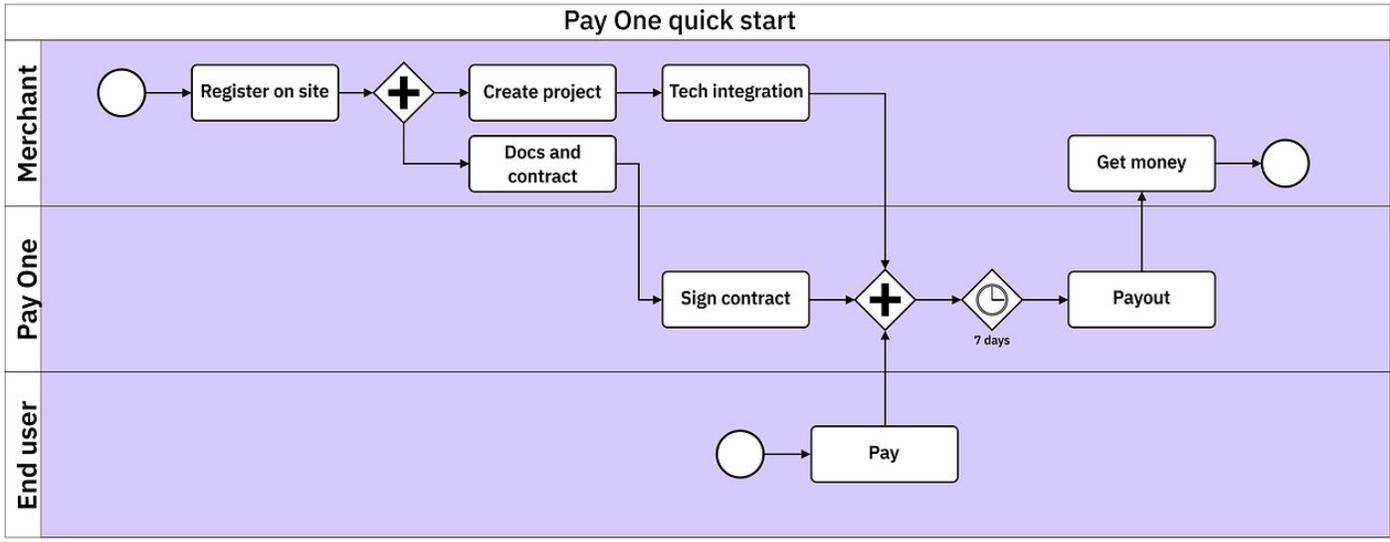
# Getting Started

Pay One is a payment system with an open source code. It accepts end-user payments, provides reports, analysis and direct access to payment data, and handles payouts. This payment solution can be either used in combination with other Protocol One components, or integrated into any third-party storefront or web page that must accept payments.

Pay One is an SaaS solution operated by Protocol One. A similar payment solution can be deployed on your own servers or in the cloud using the [Pay One open source code](#).

To get started, you need to register on the Pay One website, go through the KYC procedure and sign a contract (you can use an e-signature), then set up a project and perform technical integration on the side of your website or service. This will be enough to provide end users with a way to pay for games and other digital products.

Pay One payouts are made on a weekly basis. In most cases, the first money will be transferred to your current account within 7 days after signing the contract and accepting the first payment.



## Integration

You can start accepting payments through Pay One on your website after a few simple steps. In the case of basic payment acceptance, the integration process takes just a few minutes.



Steps of the integration process:

- **Pay One website signup.** To begin, we need only basic information about your company: name, jurisdiction, website address. You can get started immediately and use Pay One in test mode. API access keys will be generated automatically and can be changed later. Along with the next steps, you'll need to go through the KYC procedure and sign the contract with an e-signature.
- **Project configuration.** This includes configuring technical parameters of the payment interface integration and choosing one of the payment acceptance options. [Further on](#) we will examine in detail what Pay One projects are and why they are necessary, as well as talk about advanced project settings, such as product management.
- **Payment interface integration.** You can choose one of the integration methods, depending on the technologies you use and your preferences. In the basic version, you will need to add only one line to the page code. To integrate server interaction with Pay One, you can use one of the Pay One SDKs available on [GitHub](#).
- **Webhook settings.** Pay One can send client and server notifications about the transaction progress and user actions in the form. You can configure non-standard events, as well as the processing of such events on your side, if necessary.
- **Testing.** Technical integration testing is fully automatic. In a special interface, you can run a set of integration tests and get detailed results for each: description, expected result, passed or not, query text, answer text.
- **Launch.** If all tests are passed, then you only need to press one button to start.

## Accepting a Payment

For the end user, the payment process consists of clear, familiar steps, standard for most payment systems.



- **Interaction with the payment interface.** The user begins their interaction with the payment form by clicking on an interface element on the website. The form is displayed in accordance with regional features and project settings. The form responds to user actions with minimal user interface delays. In the interface, the user selects a payment method (when possible), enters plastic card information or credentials for another means of payment, passes KYC elements if necessary or fills in data required for product delivery (for example, an email or instant messenger contact to receive the key).
- **Payment confirmation.** When all the required fields in the interface are filled, the user confirms the payment and acceptance of its terms.
- **Payment status display.** The form informs on the current status of the payment. In addition, Pay One has implemented client alerts about the transaction stages, and you can implement display logic for processing them on your website.
- **Successful payment notification.** After each payment, the user receives a notification containing the transaction details and, if necessary, a key or link for using the digital product. By default, this notification is sent to the email address specified by the user (or passed on by you). The notification can also be sent to one of the instant messengers.

## Projects and Products

A single company can have an unlimited number of projects configured in Pay One. The project-level management approach allows you to independently configure Pay One for use in various contexts: different games, sites with different payment logic, specialized solutions for a particular region or a specific cohort of users.

Any information about payments in this configuration can be reflected in the reports and analytics both cumulatively and separately.

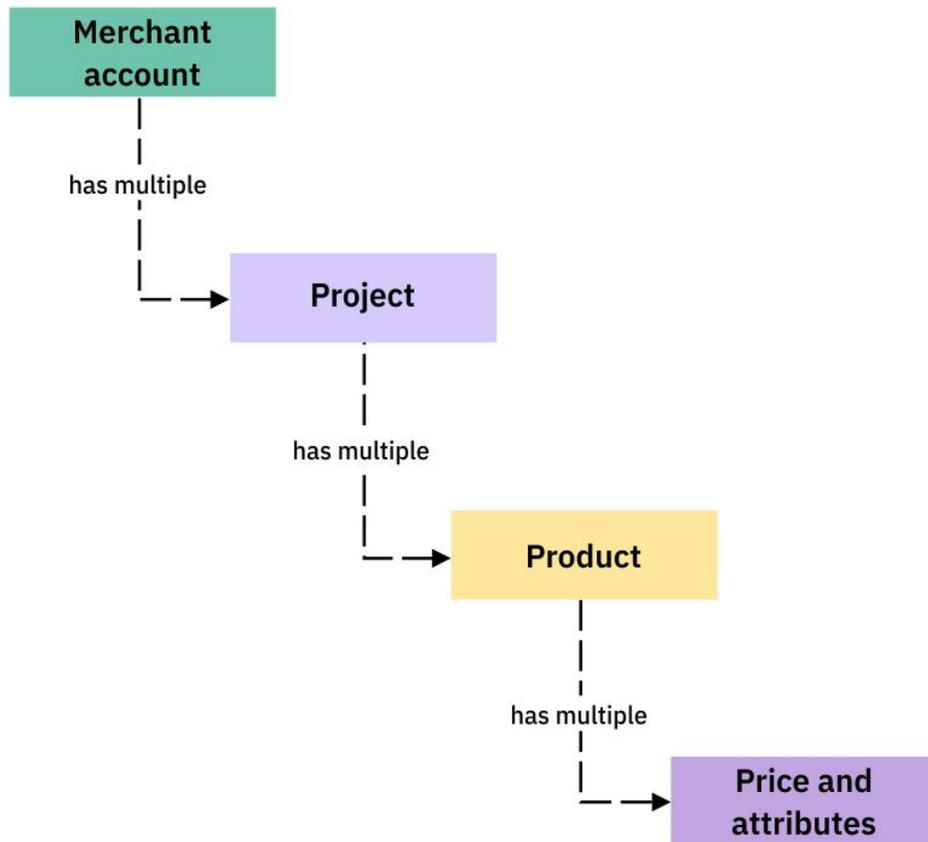
A project has two possible modes of operation:

**Simple checkout.** In this mode, you transfer the payment sum for each specific transaction to Pay One. You can select the base currency, and the payment amount for the end user will be calculated according to its exchange rate. Or you can transfer the currency each time a payment is made. This option can be useful in the following cases:

- Quick integration for a store with a small amount of products
- Test integration with a store that uses its own e-commerce engine for product management

Pay One in Simple checkout mode can send you the data on a user's region for applying regional prices from your system.

**Store.** In this mode, you manage the products and their regional prices directly within Pay One or use more advanced mechanisms of [Qilin](#) or other external service. In this case, the "one-to-many" structure is also retained within the project.



A project can contain an unlimited number of products. Each can have independently configurable regional prices or a base price in any currency. Pay One determines the user's region and applies the appropriate regional price. If a regional price is not specified, Pay One calculates the price for the end user by converting the base price into the currency of this user.

Apart from the price, a product may have a name, a short and full description, images, an icon and a text template.

Pay One supports downloading products via API or from files for synchronization with your e-commerce system.

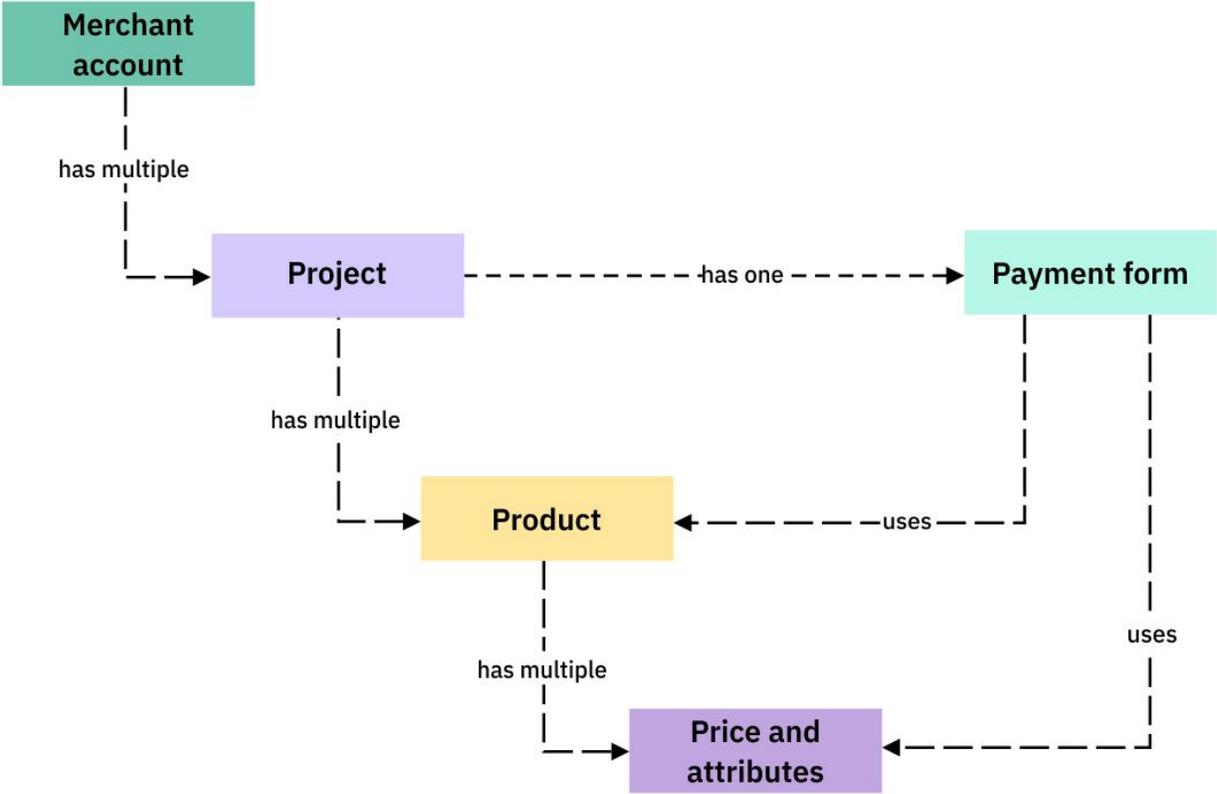
## Payment Form

Pay One payment forms are optimized for each region both in terms of the final payment methods, and at the level of UX and color gamut.

Payment forms are optimized for display on a wide range of mobile devices (tablets and smartphones). The display logic takes into account the device-specific touch controls and the use of an on-screen keyboard.

You can customize the form yourself, setting up both general properties and regional features. Pay One offers a variety of pre-made templates. The solution provides personalized recommendations for using templates based on user conversions in stores similar to yours. The longer you use Pay One, the more accurate these recommendations will be.

The payment form can be configured independently for each project.



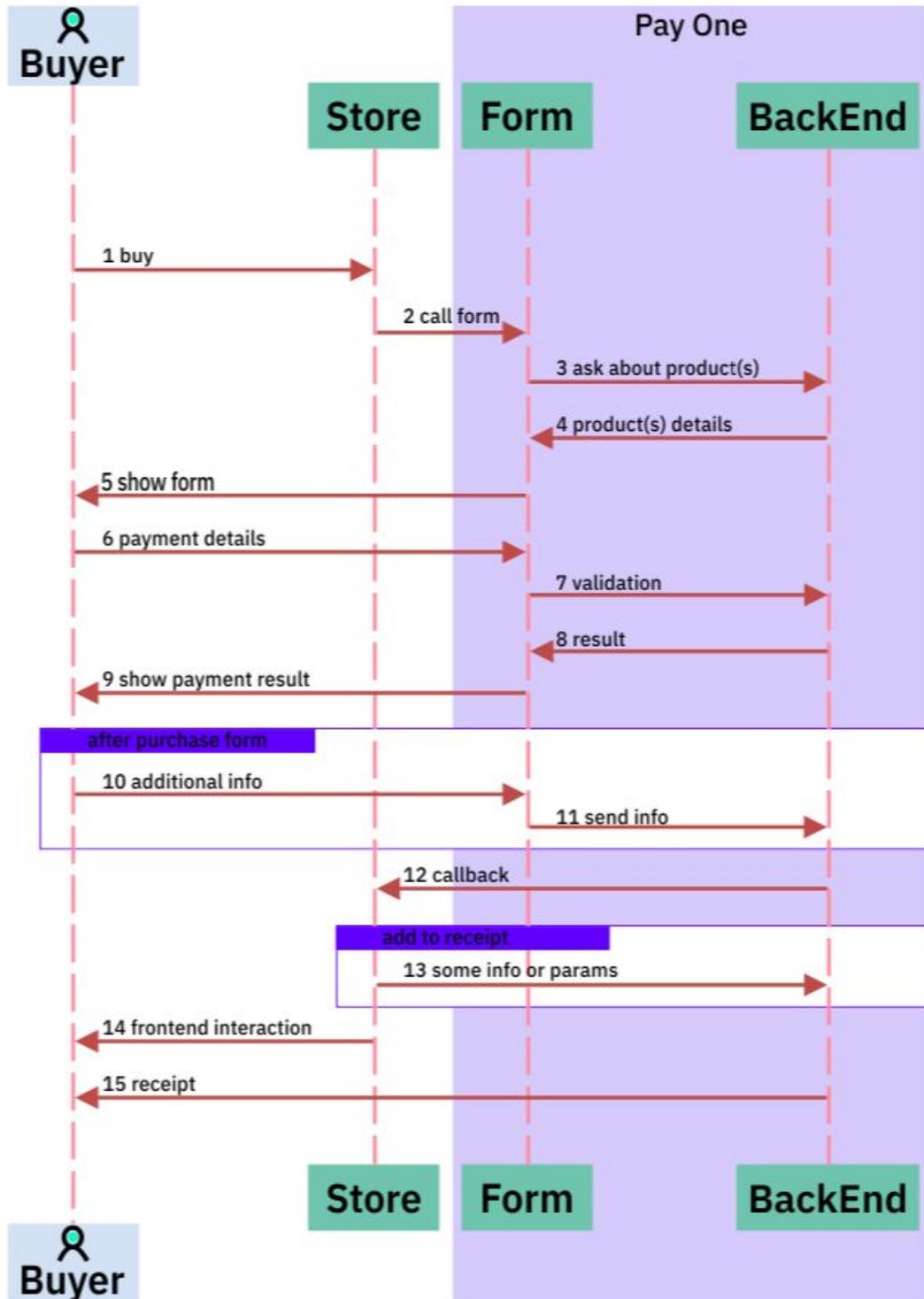
There are also two options for user identification. It can take place:

- **On the side of the Store.** The store submits to the form all user information necessary for the transaction. For example, UserID and email.
- **On the side of the Form.** The store submits to the form part of the information necessary for the transaction, and the order of steps. For example, “ask for user email on the payment success screen.” The number and type of fields and place of interaction are highly customizable.

The payment form determines the user’s country and chooses the appropriate localization language and price currency.

Elements of payment forms support event sending. You can use built-in basic mechanisms or external services, such as Google Analytics or [Uttu](#), to track conversions. Detailed conversion funnels will allow you to better analyze traffic and search for optimization points.

A common case of interaction between the user, store, form and Pay One backend is shown in the chart below:



1. The user interacts with the store page
2. The page calls the form
3. The form requests product parameters

4. The form receives product parameters
5. The payment form is shown to the user
6. The user enters details of their card or other means of payment
7. Validation on the side of Pay One
8. The form receives the Pay One server's reply
9. The form shows the user the result of the operation
10. The user fills in additional fields in the form
11. The form submits additional parameters
12. The page receives the Pay One server's reply
13. The merchant sends additional information to include in the check to the payment solution server. For instance, a key
14. The store interacts with the user in accordance with the payment results.
15. Pay One delivers the check to the user

## Embed Options

You can embed the Pay One payment form into your store or software solution in one of the following ways:

- Embedded form
- Payment opening a separate page in the browser
- Form embedded into a client app
- Form embedded into a mobile app

All embed options are equal, configured from a single interface and generate uniform events and internal analytics reports on conversions.

## Payment Methods

Pay One supports a growing number of payment acceptance methods. At this point we can process:

- Plastic cards, with or without 3D-Secure
- China Union Pay
- Neteller
- A number of cryptocurrencies: Bitcoin, Litecoin, Ethereum
- Qiwi
- Alipay
- WeChat
- Webmoney
- Bank transfers
- Payment methods of Latin America
- Payment methods of South Asia
- Payment methods of Africa

## Localization

Pay One has its own advanced algorithm for determining the region and language of the user. We do not identify them based solely on IP. We recognize a USA user who came to Germany and opened the payment interface there, so as not to make the mistake of showing them the form in German and with German payment methods.

Most of the project settings can be configured separately for each region or language:

- The list of available payment methods and their order in the payment form
- The design and order of payment form screens
- The assortment of available products
- Prices on products
- All text attributes and images of products

## Retention of Users with Unfinished Transactions

Pay One has advanced functionality to return users who for some reason did not complete the payment. When Pay One has enough data to communicate with the user and it is not prohibited by local laws, the user is prompted to finish the transaction: via email, a push-notification or instant messenger.

If integrated with [Uttu](#), you can also set up a channel for delivering such notifications through banners or other ad placements.

This approach allows you to return about 5% of users who have not completed the payment.

## Payment API

Payment API Pay One uses [REST](#) architecture. The API uses predictable, resource-oriented URLs. It uses HTTP response codes to indicate errors. The API always responds in JSON format, including in case of errors.

The API uses built-in HTTP functions, such as HTTP authentication and HTTP verbs, which can be interpreted with ready-made HTTP clients. The API supports resource sharing for different sources ([CORS](#)), allowing you to safely access it from a client web application.

All API Pay One methods support versioning and use semantic versioning (semver) for versioning management.

Payment API implements methods of working with the directory of products and payment processing that will be discussed in more detail below. It also uses an expanding set of webhooks, allowing you to receive event notifications with your Pay One transactions. You can employ webhooks to automate the backend of your store and support functions, like displaying transaction details to the end user.

In most cases, webhooks are initiated by actions on your side, but they can also be triggered by external events, such as, for instance, a user payment withdrawal notification.

To work with webhooks, you need to write and use an event handler. Examples of mechanics that you can implement on your site using webhooks:

- Unlocking an interface element for the user who has paid
- Blocking a user after payment fraud detection

Pay One provides a set of SDK libraries and plugins for interacting with the API. Currently we support SDK in the following languages: PHP, JavaScript, Go, Python, C#; and plugins for popular e-commerce solutions: Shopify, Ecwid, Wix, 3dcart, Wordpress, BigCommerce, Magento, Volusion, Big Cartel.

Payment processing can be split into two logical parts:

- Billing
- Notification of payment upon debiting funds from the user

## Payment Processing

Payment can be processed according to the typical system scenarios listed below. Sequence diagrams for each of the described scenarios can be found in [Appendix 1](#).

### Simple HTML Form

This method allows you to create an invoice by transferring a simple HTML form to the side of the payment system and is the easiest way to invoice the customer for payment.

System requests for checking order data and notifications about successful payment of an order are received routinely in accordance with the system protocol.

Basic algorithm of payment processing:

1. The buyer initiates a transaction by pressing “Buy”
2. The project collects all the necessary data from the user (in the simplest case, the project does not collect anything, but sends a pre-generated HTML form) about the buyer, generates an HTML payment form and sends the data to the system.

3. The system checks incoming data, and, if necessary, forms a server-to-server request for checking the order data on the side of the project (if specified in the technical settings of the project).
4. If the order data is not enough to make a payment, the system prompts the user to enter all the missing data into the form.
5. The system transfers the data to the payment system for payment processing.
  - a. The payment system prompts the user for confirmation of the charge-off.
  - b. The payment system notifies the system about the successful debiting of the payer.
6. The system forms a request to the project about a successful payment against an invoice.
7. The system redirects the payer to the project website page with information about the payment completion. (**Optional. If this option is not selected in the technical integration settings, then the user will be shown payment completion information on the system side.**)

## Calling JSON API

This method allows you to accept a user payment, while minimizing the need to leave the project website. It also allows maximum control over all actions of the Project user, providing a full analysis of the project user actions.

System requests for checking order data and notifications about successful payment of an order are received routinely in accordance with the system protocol.

Basic algorithm of payment processing:

1. The buyer initiates a transaction by pressing “Buy”.
2. The project processes the information received from the buyer in the previous steps and checks its completeness for invoicing according to the system protocol.
  - a. If the information received from the buyer earlier is not enough to issue an invoice according to the system protocol, then the merchant requests additional information from the buyer.
3. The project sends a server to server request to the system in accordance with the API.
4. The system processes the request received from the project, and, if necessary, sends an invoice creation request to the payment system. The project receives information on the created invoice, which contains a link to the payment confirmation form for the buyer.
5. The project redirects the buyer to the received link. (In some cases, the indicated redirection can be carried out in the iframe on the side of the Project, which allows the user not to leave the project website. Details on payment systems that support this method of integration are available upon request.)
  - a. The payment system prompts the buyer for confirmation of the charge-off.
  - b. The payment system notifies the system about the successful debiting of the payer.

6. The system forms a request to the project about a successful payment against an invoice.
7. The system redirects the payer to the project website page with information about the payment completion. (**Optional. If this option is not selected in the technical integration settings, then the user will be shown payment completion information on the system side.**)

## Payment Widget

This payment method allows you to process payments on the project website, when all the necessary payment interfaces are displayed in the payment widget. The user leaves the project website only to confirm the payment on the side of the payment system. That way, the visual presence of the system in the process is unnoticeable to the user.

System requests for checking order data and notifications about successful payment of an order are received routinely in accordance with the system protocol.

Basic algorithm of payment processing:

1. The project enables the JavaScript code of the payment widget on its website.
2. The buyer initiates a transaction by pressing "Buy".
3. The project checks the available payment information and, if necessary, records it in its accounting system.
4. The user is provided with a payment widget to fill in all the data required for making a payment. (The Payment widget can be opened both as a popup window and as an iframe on the project website. The type of opening is configured when the widget is initialized in step 1.)
  - a. When a payment widget is opened, the System receives a background request for billing and sending all the data that needs to be displayed in the payment widget.
5. The payer fills in all the necessary information in the payment widget and presses the "Pay" button.
  - a. A system call for payment fixation is made in the background.
  - b. The system, in turn, sends the data to the payment system to create a payment transaction, and the payment system returns a link to the payment verification form.
  - c. The system returns the link to the payment verification form received in the previous step to the payment widget.
  - d. The payment widget redirects the payer to the payment verification form or displays this form within itself (depending on the technical capabilities of the payment system).
6. The buyer verifies the payment.
7. The payment system notifies the system about the successful debiting of the payer.

8. The system forms a request to the project about a successful payment against an invoice.
9. The system sends the payment success information to the payment widget for displaying to the payer.

## Settlement of Disputes

A return of funds can be initiated not only by you (refund), but also on the side of the payment method, for example, at the user's request to the card-issuing bank (chargeback).

The rules for handling such situations differ depending on the payment method and territory. In all cases, Pay One consistently notifies you via the API and in your account.

As a rule, you have about 30 days to dispute a chargeback on a specific transaction by providing proof that the buyer has received the product. Pay One controls the timings and takes on the transfer of your proof to a specific payment operator. You will be able to perform all the necessary actions for this in your Pay One account.

## Personal Account

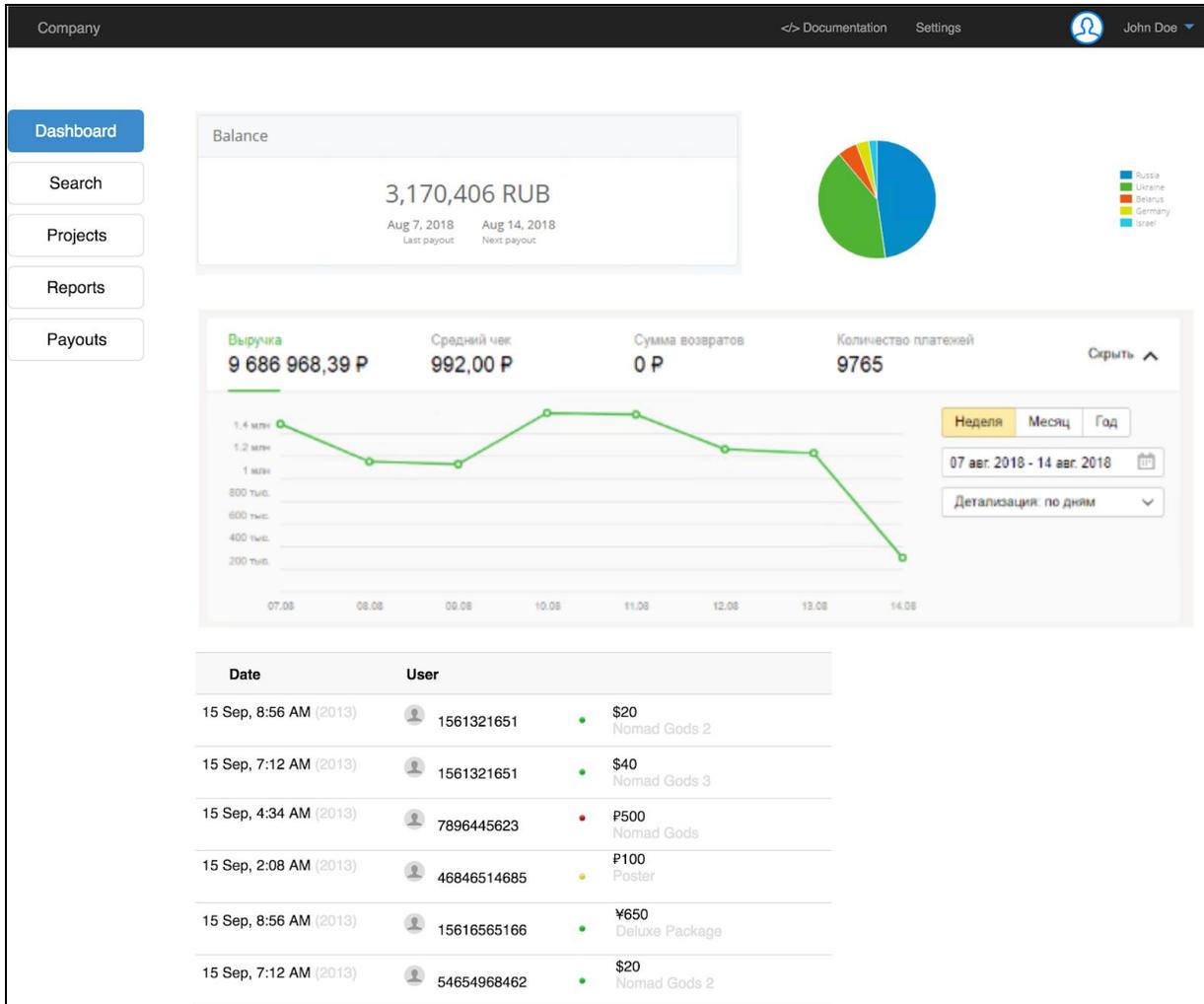
An account with Pay One is the main point of interaction with Pay One for all your employees. Pay One account contains all the necessary functions for managing and monitoring payments.

The account contains tools for:

- Account settings
- Project settings, including the launch of integration tests and product management
- Webhook configuration
- Transaction search, including full-text search for all attributes
- Returns management
- Dispute management
- Viewing reports and statistics
- Viewing and managing payouts
- Managing the rights of account users

The account supports:

- Two-factor user authentication
- Real-time data updates
- Exporting all reports in CSV and XLSX formats



Examples of reports available on the dashboard and in the Reports section:

- **Recent transactions**
- **Revenue dynamics** that can be broken down by project or product, with different levels of detail (day, week, month, year)
- **Country statistics** in the form of a table or heat map, where you can choose a period, projects and products
- **Returns report**
- **Payouts report** and a transactional report on a specific payout
- **Payment methods report**, where you can choose a period, projects and products
- **Report on conversions** from the opening of the payment form to the subsequent screens and into payment
- **Fraud report** with detailed reasons and the possibility to take over the risk of accepting a payment

# Data API

When internal reports are not enough, you can connect an external analytics tool (for example, [tableau](#) or [slemma](#)) or your own data visualization tool. You can access the data or, using the API, upload data to your storage. We adhere to the OLAP cube structure.

Dimensions:

- Project
- Product
- Geography
- Payment currency
- Payment method
- Payment date
- Return date
- User agent

And measures:

- The sum in payment currency
- The sum in payout currency
- Number of transactions

## Integration with Other Elements of Protocol One

Pay One has built-in tools for in-depth integration with other elements of the Protocol One ecosystem.

**Uttu.** For a more extensive analysis of user actions on the payment form and building end-to-end funnels, you can connect Uttu analytics tools to the website. Connecting Uttu to Pay One forms is done with just a few clicks. With Uttu, you can also manage ad targeting.

**Qilin.** If you use Qilin, it can be closely integrated with Pay One. Pay One is able to receive information about products, prices, and discounts (including personal ones) from Qilin. A key stream can be connected to the product.

**Storefront constructor.** Using the Storefront constructor, you can connect the Pay One payment form as a plugin. Pay One landing pages and payment form are optimized for collaboration.

**Auth One.** With the User Accounts Protocol One module, you can configure single sign-on authentication for your users in the payment form without writing code.

## Open Source Code

All Pay One elements have open source code published on the [GitHub](#) platform. Anyone can use the Pay One toolkit to create their own payment solution. This will require you to:

- Download and deploy the necessary Pay One elements in your infrastructure
- Sign a contract for accepting payments with a bank (or banks)
- Pass [PCI DSS](#) certification

Open source allows us to implement best practices developed by the community in our SaaS-solution in the shortest possible time. The best versions of customized forms, which take into account the specifics of use and legislation in different regions, are constantly added to the Pay One payment form template library.

## Disclaimer

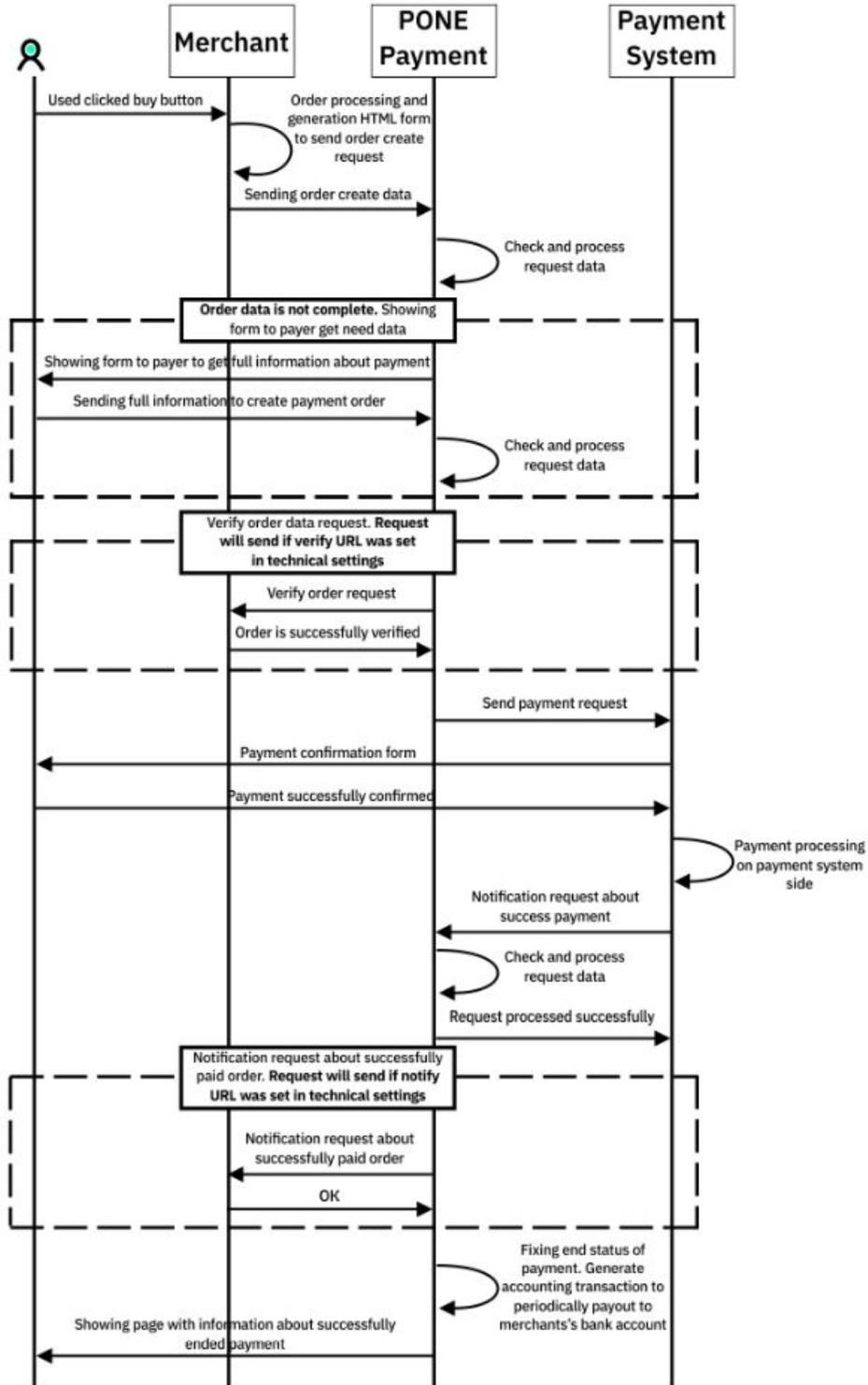
This document does not represent a solicitation of investment or any other form of material support for the Protocol One project, Pay One or the Open Game Data Exchange. The purpose of this document is to provide a detailed and comprehensive description of Pay One, approaches and principles for accepting payments from buyers of games and other digital assets.

Statements and other information of a declarative nature contained within this document must not be construed as direct assertions or promises unless they are expressly specified as such.

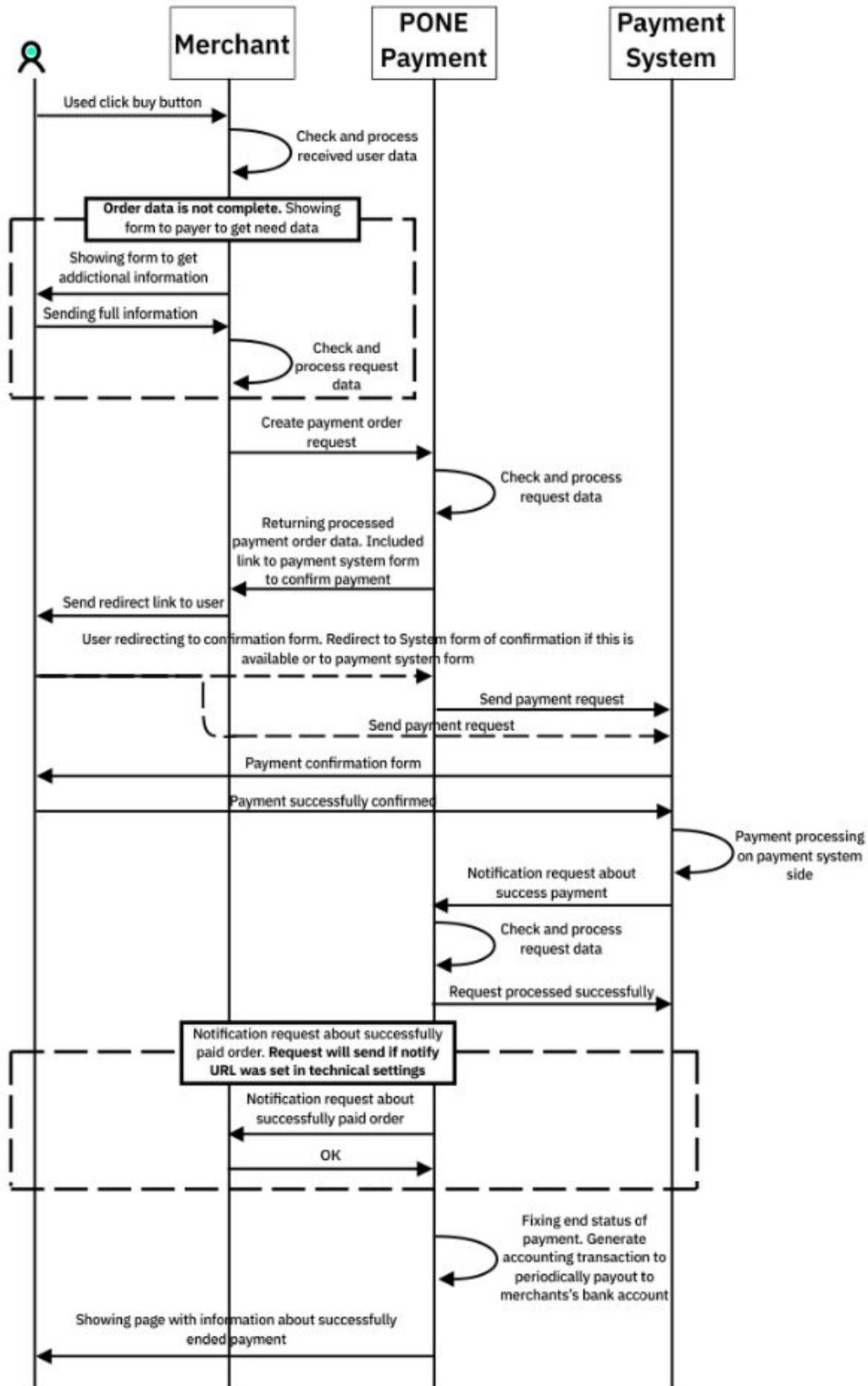
In the current edition, we actively collect feedback from the video game developers and publishers, stores and platforms to improve the specifications and make the product we create easier and more convenient for all participants.

# Appendix 1. Payment Processing Charts

HTML form:



JSON API:



Payment widget:

