

PayPal Workflow Connector

Download the PDF of this article.

In this Article

- [Overview](#)
- [Requirements](#)
- [General Connector Configuration](#)
- [Configuring PayPal Transactions](#)
- [Checkout Flow](#)
- [Error Handling](#)
- [Testing and Best Practices](#)
- [Frequently Asked Questions](#)

Related Articles

Overview

The PayPal Workflow Connector enables you to process payments directly within your FormAssembly workflows. This guide will help you configure and use the connector to accept one-time payments through PayPal.

Requirements

To use the PayPal Workflow Connector, a FormAssembly user must

- purchase FormAssembly's Payment Connector Add-on with PayPal user permission enabled
- have a PayPal Business or Personal Account (Production or Sandbox)

Supported Payment Types

- **One-time charges:** Accept single payments with support for multiple line items

General Connector Configuration

Refer to our [Workflow-native Connectors article](#) for general configuration information.

Note: When adding a PayPal Workflow Connector, a Page Redirect Step will **automatically** be added after the PayPal step. The Page Redirect Step is needed to avoid potential workflow processing errors.

PayPal Account Authorization

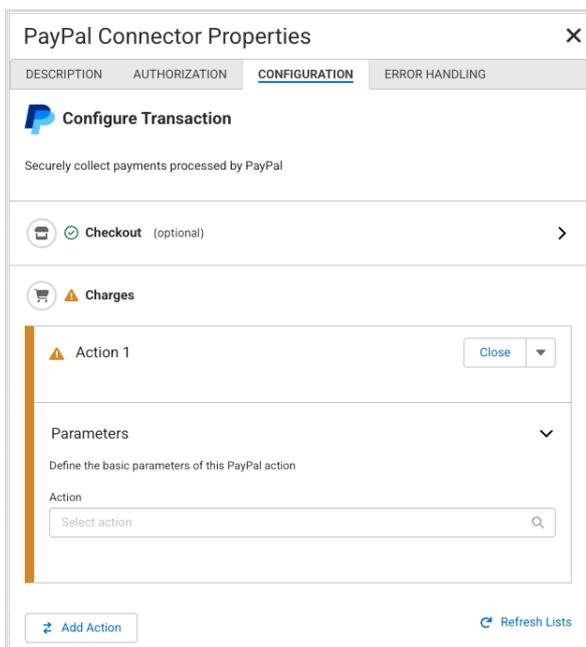
When completing your PayPal authorization on the Authorization tab, select the appropriate environment for the credentials you will use for authentication.

- **Sandbox:** For testing (uses PayPal Sandbox credentials)
- **Production:** For live payments (uses PayPal Production credentials)

Note: The connector uses OAuth 2.0 for secure authentication. Your credentials are encrypted, and tokens are automatically refreshed.

Configuring PayPal Transactions

The Configuration tab for this connector has two main sections: **Charges** and **Checkout**. The Charges section is where you define what the transaction is (what you are charging for). The Checkout section is used for mapping transaction details and configuring variables to be returned.



Charges Section Configuration

The **Set Up One-Time Charge** and **Add One-Time Charge Line Item** actions work as a parent/child pair to enable single-charge (non-recurring) transactions. These actions must have a **dependency** set between them to function correctly.

Action: Set Up One-Time Charge

The **Set Up One-Time Charge** action creates the base one-time charge transaction. This action defines the parameters required to initialize the checkout session for one-time charges.

Dependent **Add One-Time Charge Line Item** actions can be added to specify individual products or charges. Each dependent line item must reference the parent **Set Up One-Time Charge** action through its dependency settings. When a one-time charge action is added, a corresponding dependent line item action is automatically added.

Action: Add One-Time Charge Line Item

This dependent action defines the individual line item(s) for the one-time charge checkout. Define a custom charge line item with the following fields:

- **Name:** (Required) The name of the product or service
- **Description:** Additional details about the item
- **Item ID:** The internal reference code for the item
- **Currency:** (Required) The currency used for the transaction
 - **Note:** The currency selected must be the **same for all line items** of a charge if multiple line items are added to a charge action.
- **Price:** (Required) Cost per unit
- **Quantity:** (Required) Number of units

The Price and Quantity fields can be mapped with the following value types:

- **Form Field:** data entered into a form field by a respondent
- **Value or Formula:** static text, aliases, formulas, or calculations
- **Variable:** custom variables created in other Workflow-native Connectors

Multiple One-Time Charges

Multiple one-time charges can be added to a single checkout by selecting **Add Action** → **Add One-Time Charge Line Item** for each additional line item. Each action must have a dependency set to a parent **Set Up One-Time Charge** action.

Alternatively, you can select **Add Line Item** within the **Set Up One-Time Charge** action to add more line items.

When a respondent initiates a checkout, all line items will be combined into a single PayPal order within the one checkout session.

Checkout Section Configuration (optional)

The **Checkout** section contains subsections for transaction data mapping (**Map**) and **Return Variables**.

Transaction Data Mapping

Select **Add Mapping**, within the Map subsection of Checkout, to map values for **Custom ID** and **Soft Descriptor** to the transaction.

- **Custom ID:** Allows you to associate FormAssembly data with PayPal transactions. When you map a value to the Custom ID field, it appears in the PayPal transaction details, helping you track and identify specific transactions.
 - **Note:** This can be used to add a “customer note” by mapping a form field to the Custom ID field.
- **Soft Descriptor:** Allows you to customize how the transaction appears on the customer’s credit card or bank statement. The mapped value will display alongside your business name in the transaction description in the following format:
 - Format = PAYPAL + <BUSINESS NAME> + <SOFT DESCRIPTOR>
 - **Note:** The transaction description is limited to 22 characters.

Return Variables

Return variables can be mapped to form fields or workflow variables, allowing payment details to be referenced in later workflow steps or for confirmation messages. The following values from a one-time charge transaction can be set as return variables for use later in the workflow:

VARIABLE	DESCRIPTION	USE CASE
Transaction ID	Unique PayPal transaction identifier	Order tracking, receipts
Transaction URL	Link to view the transaction in PayPal	Customer support, verification
Transaction Amount	Total amount charged	Confirmation emails, reporting
Transaction Status	Payment status (completed, pending, failed)	Conditional routing
Payment Source	Always returns "PayPal"	Analytics
Customer Email	Payer's email address	Communication, CRM integration
Payer ID	PayPal payer identifier. (Note: Customer emails can change over time, but their Payer ID is always the same)	Customer tracking

Checkout Flow

What Respondents Experience

1. **Form Submission:** Respondent completes and submits the form
2. **Automatic Redirect:** They are redirected to PayPal's secure checkout page
3. **Payment:** They complete payment using PayPal with a card or another configured payment option
4. **Return to Workflow:** They are automatically returned to your workflow

Possible Outcomes

- **Successful Payment:** Workflow continues with return variables populated if configured
- **Checkout Cancellation:** Respondent cancels and returns to workflow (configurable routing)
- **Payment Error:** Error is logged, and workflow follows configured error handling rules

Error Handling

If an error occurs during the checkout process—or if the respondent cancels the payment by clicking the back button—the workflow will follow the routing defined in the **Error Handling** tab.

The Error Handling configuration includes a field to specify which step the respondent should be directed to in case of an error. This applies to both payment failures and user-initiated cancellations.

Common Issues and Solutions

ISSUE	POSSIBLE CAUSE	SOLUTION
Authentication fails	Invalid credentials or expired token	Re-authenticate your PayPal account
Payment declined	Insufficient funds or card issues	Respondent should try another payment method
Checkout timeout	Respondent took too long	No solution - Handle with error handling configuration
API error	PayPal service issue	Check the PayPal status page; retry later

Testing and Best Practices

Using Sandbox Mode

1. Use a PayPal Sandbox account at developer.paypal.com
2. Authenticate the connector within FormAssembly in Sandbox mode with sandbox credentials on the testing tools/Sandbox Account page in PayPal
3. Verify intended configurations are populated correctly (line items, transaction data, return variables, etc.)
4. Test error scenarios (declined payments, cancellations)

Before Going Live

- Test all line item configurations
- Verify return variable mapping
- Test error handling based on payment outcome
- Review error handling behavior
- Switch to Production environment
- Process a small test transaction with a real payment

Best Practices

Security

- Always use Production mode for live payments
- Never share your PayPal credentials
- Regularly review transaction logs
- Enable two-factor authentication on your PayPal account

User Experience

- Provide clear product descriptions
- Set accurate prices and quantities
- Use meaningful item codes for tracking
- Test the complete checkout flow regularly

Workflow Design

- Implement conditional routing based on payment status
- Log transaction IDs for customer support

Frequently Asked Questions

Can I accept payments in currencies other than USD?

Currency support depends on your PayPal account settings. Configure multi-currency in your PayPal Business Account.

Why should I upgrade from the Form-native PayPal connector to the PayPal Workflow-native connector?

The Form-native PayPal connector uses Website Payments Standard (WPS), which PayPal is retiring on January 1, 2027. To ensure uninterrupted payment processing, it is strongly recommended that you upgrade to the PayPal Workflow-native connector, which supports PayPal Checkout.

Source: <https://developer.paypal.com/limited-release/multiparty/wps-deprecation/>
