





# Migration Example

Platforms	Web
Products	<ul style="list-style-type: none"><li>Payments: Best Practices, PayPal Checkout</li></ul>
Products Options	<ul style="list-style-type: none"><li>Checkout Methods: Authorize</li></ul>
Revision History	<ul style="list-style-type: none"><li><div>v1.0</div><div>Jan 28, 2024</div><div>Initial Version</div></li></ul>
Introduction	Integration Description

Legal Notice

© 2023 PayPal, Inc. All rights reserved. PayPal is a registered trademark of PayPal, Inc. The PayPal logo is a trademark of PayPal, Inc. Other trademarks and brands are the property of their respective owners. The information in this document belongs to PayPal, Inc. It may not be used, reproduced or disclosed without the written approval of PayPal, Inc. Notice of non-liability: PayPal, Inc. is providing the information in this document to you “AS-IS” with all faults. PayPal, Inc. makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. PayPal, Inc. assumes no liability for damages (whether direct or indirect), caused by errors or omissions, or resulting from the use of this document or the information contained in this document or resulting from the application or use of the product or service described herein. PayPal, Inc. reserves the right to make changes to any information herein without further notice.

# Table of Contents

## 1. Payments

---

- 1.1. PayPal Commerce Platform (PPCP)
  - 1.1.1. Features
  - 1.1.2. Frontend and Backend
  - 1.1.3. API Flows
  - 1.1.4. Parity
  - 1.1.5. API Calls
    - 1.1.5.1. Overview
    - 1.1.5.2. Documentation
    - 1.1.5.3. Create an Order
    - 1.1.5.4. Update an Order
    - 1.1.5.5. Capture Order Request
    - 1.1.5.6. Show Order Details
    - 1.1.5.7. Authorize and Capture
    - 1.1.5.8. Void an authorization
  - 1.1.6. Legal Implications
  - 1.1.7. Alternative additional placements
  - 1.1.8. Payment Page – PayPal Exclusives
  - 1.1.9. PayPal Review page

# 1. Payments

---

## 1.1. PayPal Commerce Platform (PPCP)

### 1.1.1. Features

The **PAYPAL COMMERCE PLATFORM (PPCP)** is PayPal's newest payment processing API suite. The **PPCP** API suite encompasses the following payment methods:

- PayPal (branded wallet experience)
- Pay Later

### 1.1.2. Frontend and Backend

---

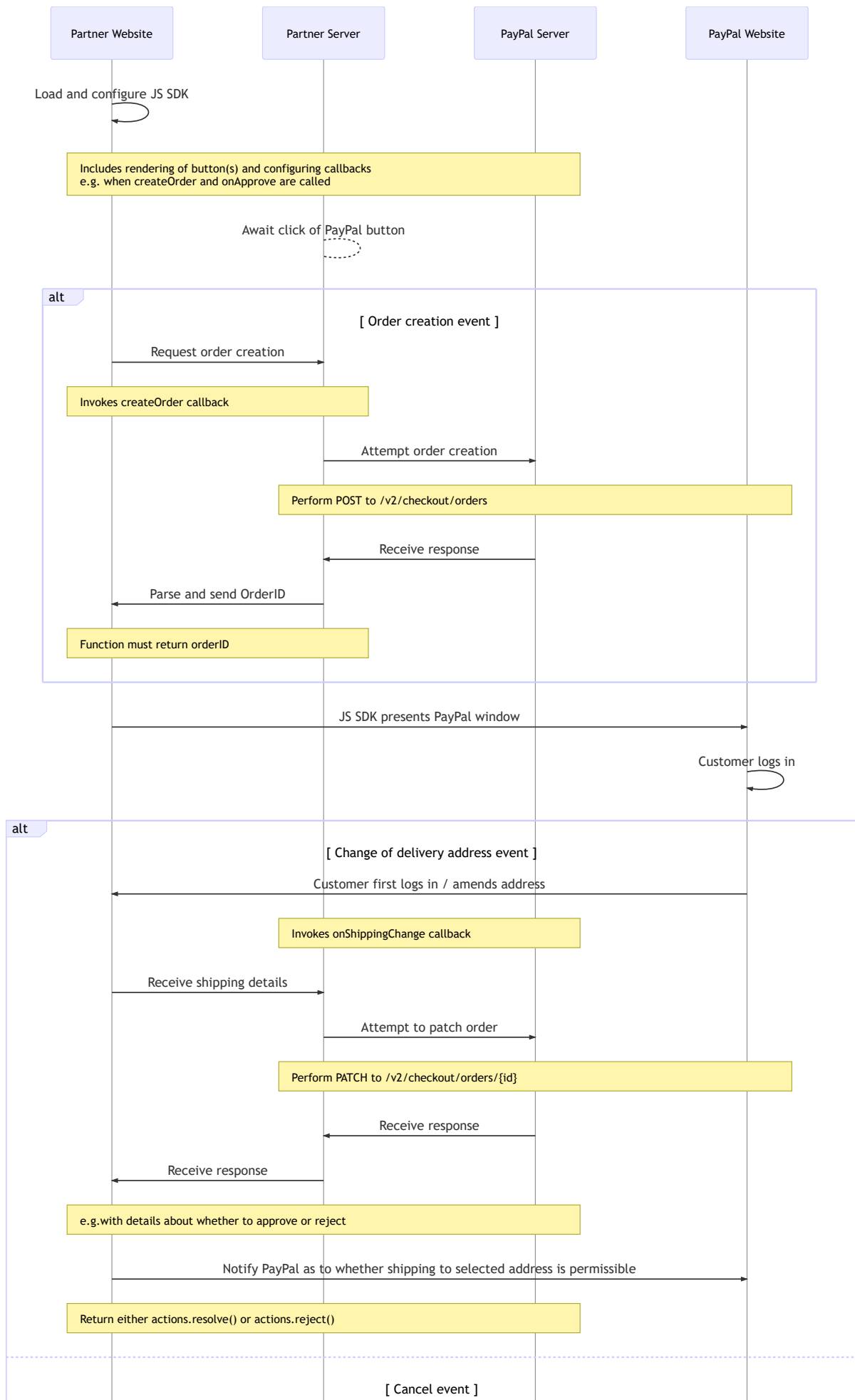
As opposed to only offering a backend API as PayPal did in previous API suites, e.g. the **EXPRESS CHECKOUT / CLASSIC API**, **PPCP** offers both a backend API and a frontend library, called JavaScript SDK (**JS SDK**).

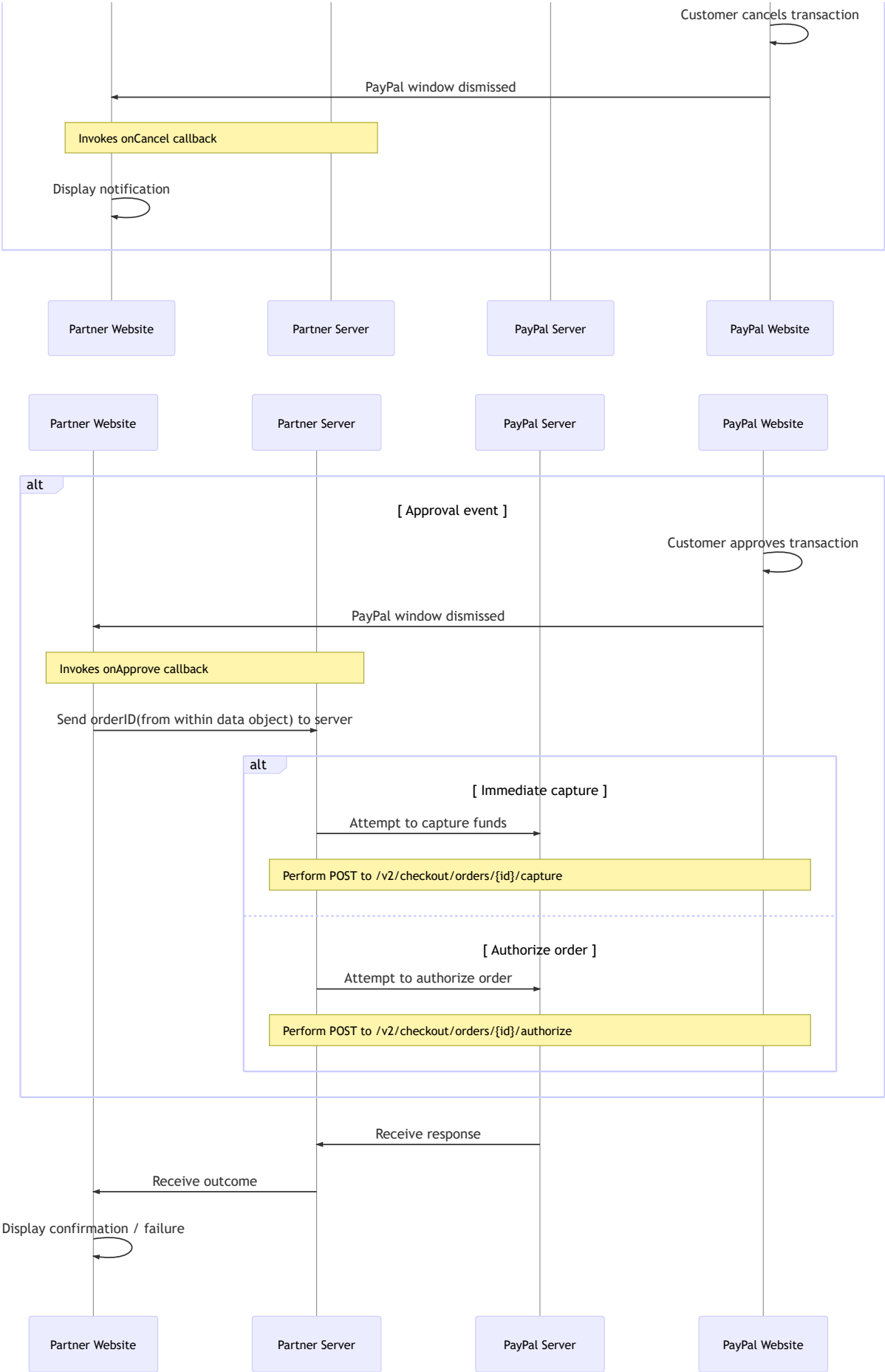
A **PPCP** integration will thus always involve two items:

- Backend API, called **ORDERS v2**
- Frontend SDK, called JavaScript SDK (**JS SDK**)

## 1.1.3. API Flows

---





## 1.1.4. Parity

---

Especially when other payment methods are offered earlier in the checkout journey, it is required to show PayPal on these same pages, called **PARITY**.

## 1.1.5. API Calls

---

### 1.1.5.1. Overview

To process payments, you will need to proceed with the following phases:

1. **Create an order:** An order represents the payment you want to process and includes details, such as the amount to process, details about the seller, and any fees you want to collect.
2. **Render payment methods:** Present payment methods to your buyers so they can pay for the order. Smart Payment Buttons allows you to present several payment methods including PayPal, Pay Later, Credit & Debit Cards, Venmo, iDEAL, Bancontact, and Sofort.
3. **Move money:** Once your buyer has approved for the order, you can move money from your buyer to your seller. To move money you **must capture** it from your buyer. With immediate capture, you capture money as soon as your buyer pays for the order.

### 1.1.5.2. Documentation

Please find the official documentation for all API parameters here:

<https://developer.paypal.com/docs/api/orders/v2/>.

### 1.1.5.3. Create an Order

Seller must pass line-item details in the Order v2 APIs (Create Order > Purchase Units > Items)

Seller must specify if the item is Physical/Tangible: (Create Order > Purchase Units > Items > Category: PHYSICAL\_GOODS). Refer to

<https://developer.paypal.com/docs/api/orders/v2/#definition-item> for further details.

Seller must pass Shipping Address for the Physical/Tangible Goods in the Order v2 APIs. Refer to [https://developer.paypal.com/docs/api/orders/v2/#definition-shipping\\_detail](https://developer.paypal.com/docs/api/orders/v2/#definition-shipping_detail) for further details.



### 1.1.5.3.1. Request

```
POST https://api-m.paypal.com/v2/checkout/orders/  
Content-Type: application/json  
PayPal-Request-Id: UNIQUE_CALL_ID  
Authorization: Bearer MERCHANT_ACCESS_TOKEN  
PayPal-Partner-Attribution-Id: BN_CODE
```

```

{
  "intent": "CAPTURE",
  "purchase_units": [
    {
      "reference_id": "d9f80740-38f0-11e8-b467-0ed5f89f718b",
      "amount": {
        "currency_code": "USD",
        "value": "204.00",
        "breakdown": {
          "item_total": {
            "currency_code": "USD",
            "value": "199.00"
          },
          "shipping": {
            "currency_code": "USD",
            "value": "5.00"
          }
        },
      },
      "items": [
        {
          "name": "DSLR Lens Cover",
          "sku": "517293",
          "unit_amount": {
            "currency_code": "USD",
            "value": "204.0"
          },
          "tax": {
            "currency_code": "USD",
            "value": "0.0"
          },
          "quantity": "1",
          "category": "PHYSICAL_GOODS"
        }
      ],
      "shipping": {
        "name": {
          "full_name": "Frank Geller",
          "shipping_type": "SHIPPING"
        },
        "address": {
          "address_line_1": "4668-xxxx",
          "admin_area_2": "Brownsville",
          "admin_area_1": "TX",
          "postal_code": "785-xxxx",
          "country_code": "US"
        }
      }
    }
  ],
  "payment_instruction": {
    "disbursement_mode": "INSTANT"
  },
  "payment_source": {
    "paypal": {
      "experience_context": {
        "payment_method_preference": "IMMEDIATE_PAYMENT_REQUIRED",
        "payment_method_selected": "PAYPAL",
        "brand_name": "EXAMPLE INC",
        "locale": "en-US",
        "landing_page": "LOGIN",
        "shipping_preference": "SET_PROVIDED_ADDRESS",
        "user_action": "PAY_NOW",
        "return_url": "https://example.com/returnUrl",
        "cancel_url": "https://example.com/cancelUrl"
      }
    }
  }
}

```

```

    }
  }
}

```

### 1.1.5.3.2. Response

HTTP 200

```

{
  "id": "2AC63827BP9351315",
  "status": "PAYER_ACTION_REQUIRED",
  "payment_source": {
    "paypal": {}
  },
  "links": [
    {
      "href": "https://api.sandbox.paypal.com/v2/checkout/orders/2AC63827BP9351315",
      "rel": "self",
      "method": "GET"
    },
    {
      "href": "https://www.sandbox.paypal.com/checkoutnow?token=2AC63827BP9351315",
      "rel": "payer-action",
      "method": "GET"
    }
  ]
}

```

### 1.1.5.4. Update an Order

Updates an order that has the CREATED or APPROVED status. You cannot update an order with COMPLETED status. Details can be found at [https://developer.paypal.com/docs/api/orders/v2/#orders\\_patch](https://developer.paypal.com/docs/api/orders/v2/#orders_patch)

You can patch these attributes and objects:

Attribute	Op
purchase_units	replace, add
purchase_units[].custom_id	replace, add, remove
purchase_units[].description	replace, add, remove
purchase_units[].payee.email	replace, add
purchase_units[].shipping	replace, add, remove
purchase_units[].soft_descriptor	replace, add, remove
purchase_units[].amount	replace
purchase_units[].invoice_id	replace, add, remove

To make an update, you must provide a `reference_id`. If you omit this value with an order that contains only one purchase unit, PayPal sets the value to default which enables you to use the path: `"/purchase_units/@reference_id=='default'/{attribute-or-object}"`.

```
PATCH https://api-m.paypal.com/v2/checkout/orders/{order_id}
```

```
Content-Type: application/json
```

```
PayPal-Request-Id: UNIQUE_CALL_ID
```

```
Authorization: Bearer MERCHANT.ACCESS_TOKEN
```

```
PayPal-Partner-Attribution-Id: BN_CODE
```

```
[
  {
    "op": "add",
    "path": "/purchase_units/@reference_id=='default'/invoice_id",
    "value": "03012022-3303-01"
  }
]
```

### 1.1.5.5. Capture Order Request

Details can be found at [https://developer.paypal.com/docs/api/orders/v2/#orders\\_capture](https://developer.paypal.com/docs/api/orders/v2/#orders_capture)

```
POST https://api-m.paypal.com/v2/checkout/orders/TRANSACTION.ORDER_ID/capture
```

```
Content-Type: application/json
```

```
PayPal-Request-Id: UNIQUE_CALL_ID
```

```
Authorization: Bearer MERCHANT.ACCESS_TOKEN
```

```
PayPal-Partner-Attribution-Id: BN_CODE
```

-- No body

### 1.1.5.5.1. Response

```
{
  "id": "50190127TN364715T",
  "status": "COMPLETED",
  "payment_source": {
    "paypal": {
      "email_address": "jchapman05@paypal.com",
      "account_id": "ZGU4HHWUDTMD2",
      "name": {
        "given_name": "John",
        "surname": "Chapman"
      },
      "address": {
        "country_code": "US"
      }
    }
  },
  "payer": {
    "name": {
      "given_name": "John",
      "surname": "Doe"
    },
    "email_address": "customer@example.com",
    "payer_id": "QYR5Z8XDVJNXQ"
  },
  "purchase_units": [
    {
      "reference_id": "d9f80740-38f0-11e8-b467-0ed5f89f718b",
      "shipping": {
        "address": {
          "address_line_1": "2211 N First Street",
          "address_line_2": "Building 17",
          "admin_area_2": "San Jose",
          "admin_area_1": "CA",
          "postal_code": "95131",
          "country_code": "US"
        }
      },
      "payments": {
        "captures": [
          {
            "id": "3C679366HH908993F",
            "status": "COMPLETED",
            "amount": {
              "currency_code": "USD",
              "value": "100.00"
            },
            "seller_protection": {
              "status": "ELIGIBLE",
              "dispute_categories": [
                "ITEM_NOT_RECEIVED",
                "UNAUTHORIZED_TRANSACTION"
              ]
            },
            "final_capture": true,
            "disbursement_mode": "INSTANT",
            "seller_receivable_breakdown": {
              "gross_amount": {
                "currency_code": "USD",
                "value": "100.00"
              }
            }
          }
        ]
      }
    }
  ]
}
```

```

        "paypal_fee": {
            "currency_code": "USD",
            "value": "3.00"
        },
        "net_amount": {
            "currency_code": "USD",
            "value": "97.00"
        }
    },
    "create_time": "2018-04-01T21:20:49Z",
    "update_time": "2018-04-01T21:20:49Z",
    "links": [
        {
            "href": "https://api-
m.paypal.com/v2/payments/captures/3C679366HH908993F",
            "rel": "self",
            "method": "GET"
        },
        {
            "href": "https://api-
m.paypal.com/v2/payments/captures/3C679366HH908993F/refund",
            "rel": "refund",
            "method": "POST"
        }
    ]
}
]
}
],
"links": [
    {
        "href": "https://api-m.paypal.com/v2/checkout/orders/50190127TN364715T",
        "rel": "self",
        "method": "GET"
    }
]
}

```

#### 1.1.5.5.2. Error Handling

It will be important to check the `payments.captures.status` when you receive an HTTP 201 response and also when you receive an HTTP 422 error, so that you can display the appropriate message to the buyer.

- Pending Transactions – You would receive an HTTP 201 response in this case, but within the `payments.captures` object, you will see the status for this transaction. If it is “PENDING” then that means the capture was completed and the money was moved out of the buyer’s account, but it is being held until the pending status is cleared. In some cases, the seller may need to log into their account to take some action. In other cases, the pending status means that PayPal needs to do a review of the transaction and will automatically either approve or decline the transaction based on the review. You would receive the `PAYMENT.CAPTURE.COMPLETED` or the `PAYMENT.CAPTURE.DENIED` webhook based on the results.
- Other HTTP 422 response errors – This is where the transaction stopped processing before the capture could be completed. This can be something like a Duplicate Invoice ID error or if the Line Item amounts do not add up to the total amount. In the API response, you will see a “details” object with the name and a description of the error. In some cases, this can just be a validation error. It’s recommended to display the message from the “description” to the buyer as it gives details on what may have caused the transaction to have thrown the error.
- HTTP 201 response (but denied) – This is where you would receive an HTTP 201 result, but in the captures object you will see a status of “DENIED” or “DECLINED”. This means that the transaction was declined.

To test the scenario for a Pending transaction you can do something like this:

- Create a US based seller from within `developer.paypal.com`. Connect that seller to your platform.
- Once the account has been created, go to the Dashboard -> Sandbox -> Accounts and find the account that was just created. Under the Manage Accounts column, click on View/Edit for this Sandbox account.
- On the Account Details page, click the Settings tab.
- Toggle the Payment Review option to “On”. This will make all transactions with this seller go into the PENDING status with the reason of “PENDING\_REVIEW”.

In these cases, the seller is required to capture the order immediately after receiving the webhook. Afterwards an immediate payment confirmation can be shown on the seller's website / app.

### 1.1.5.6. Show Order Details

To see your order details, pass the order ID as a path parameter in a show order details call.

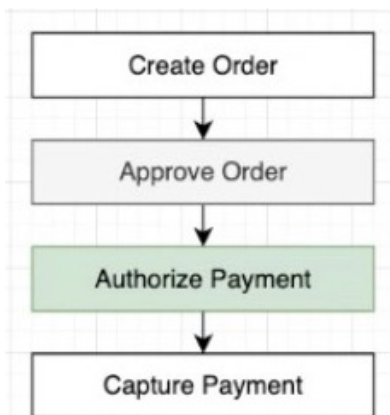
```
GET https://api-m.paypal.com/v2/checkout/orders/50190127TN364715T
Content-Type: application/json
PayPal-Request-Id: UNIQUE_CALL_ID
Authorization: Bearer MERCHANT.ACCESS_TOKEN
PayPal-Partner-Attribution-Id: BN_CODE
```



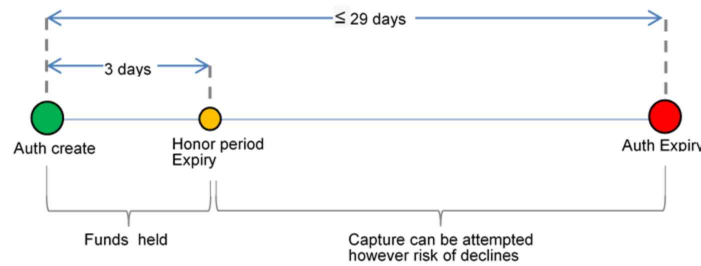
A successful request returns the HTTP `200 OK` status code and a JSON response body that shows order details.

```
{
  "id": "50190127TN364715T",
  "status": "CREATED",
  "intent": "CAPTURE",
  "purchase_units": [
    {
      "reference_id": "d9f80740-38f0-11e8-b467-0ed5f89f718b",
      "amount": {
        "currency_code": "USD",
        "value": "100.00"
      }
    }
  ],
  "create_time": "2018-04-01T21:18:49Z",
  "links": [
    {
      "href": "https://api-m.paypal.com/v2/checkout/orders/50190127TN364715T",
      "rel": "self",
      "method": "GET"
    },
    {
      "href": "https://www.paypal.com/checkoutnow?token=50190127TN364715T",
      "rel": "approve",
      "method": "GET"
    },
    {
      "href": "https://api-m.paypal.com/v2/checkout/orders/50190127TN364715T/capture",
      "rel": "capture",
      "method": "POST"
    }
  ]
}
```

### 1.1.5.7. Authorize and Capture



Authorize and capture allows you to authorize your buyers funds before you capture them. An authorization places a hold on the funds and is valid for 29 days. Please refer also to the official documentation <https://developer.paypal.com/docs/checkout/standard/customize/authorization-honor/>



After a successful authorization, PayPal recommends that you capture the funds within the three-day honor period. Within the 29-day authorization period, you can issue multiple re-authorizations after the honor period expires. A re-authorization generates a new Authorization ID and restarts the honor period and any subsequent capture should be performed on the new Authorization ID. If you do a re-authorization on the 27th day of the authorization, you get only two days of honor period.

To split authorize and capture into separate actions, add `&intent=authorize` to the PayPal JavaScript SDK script tag as seen in the following example.

```
<script src="https://www.paypal.com/sdk/js?client-id=YOUR_CLIENT_ID&intent=authorize">
</script>
```

Before you authorize funds, you must create an order with the intent field set to `AUTHORIZE`.

```
{
  "intent": "AUTHORIZE",
  "purchase_units": [{
    "amount": {
      "currency_code": "USD",
      "value": "100.00"
    },
    "payee": {
      "email_address": "seller@example.com"
    }
  }]
}
```

After your buyer approves the order, you call `authorize order` to authorize the buyer's funds.

```
POST https://api-m.paypal.com/v2/checkout/orders/50190127TN364715T/authorize
```

Then you have to capture the funds. Please see full documentation [here](#)

### 1.1.5.8. Void an authorization

VOIDs, or cancels, an authorized payment, by ID. You cannot void an authorized payment that has been fully captured.

```
POST /v2/payments/authorizations/{authorization_id}/void
```

## 1.1.6. Legal Implications

With regards to the EU Button Law it is suggested to use `style.label: "paypal"`. Note that this suggestion does not constitute legal advice and does not relieve the seller from obtaining legal advice.

## 1.1.7. Alternative additional placements

- A. Feature PayPal Checkout buttons prominently on Login / Guest checkout pages.
- B. Display PayPal Pay Later banner near price or checkout buttons

**MERCHANT** SECURE CHECKOUT

### LOG IN OR REGISTER

EMAIL ADDRESS

PASSWORD:

**LOG IN**

REGISTER

[FORGOT YOUR PASSWORD?](#)

**PayPal** Buy now, pay later. B

A **PayPal Checkout**

**Pay Later**

**venmo Checkout**

**Debit or Credit Card**

Powered by **PayPal**

### GUEST CHECKOUT

Finalize your order without creating an account.

**CONTINUE AS A GUEST**

[PRIVACY STATEMENT](#) [SHIPPING AND RETURN POLICIES](#) © 2022 ALL RIGHTS RESERVED

## 1.1.8. Payment Page – PayPal Exclusives

- A Feature PayPal prominently above all payment options on the payment page. Vertical button layout is recommended for both desktop and mobile. Button size is at parity with other payment methods.

- 26% of customers abandon at the payment stage because “my preferred payment method was not available” so present PayPal equally with Credit/Debit in radio button format.\*

**MERCHANT** SECURE CHECKOUT

TELEVISION & VIDEO CAMERA & PHOTO HEADPHONES PORTABLE AUDIO CELL PHONES & ACCESSORIES

HOME > **SECURE CHECKOUT**

**Payment**

A **PayPal**

**venmo**

☐ Pay by Card

☐ Apple Pay

☐ Google Pay

**Order Summary**

Product	Total
VANITY WALL PROJECTOR 1080P QTY:1	\$74.29
TAX	\$6.00
<b>Sub total</b>	<b>\$80.29</b>

**Shipping Method** Free Shipping Pay Later 10.00 Express shipping 122.00

**TOTAL** **\$80.39**

**REVIEW ORDER**

**Payment**

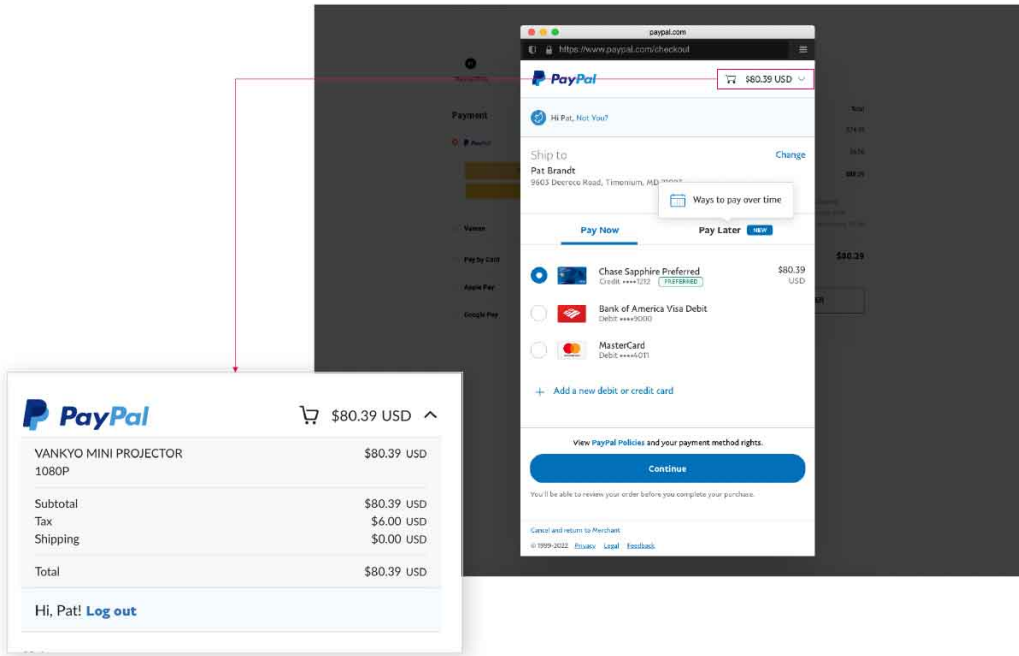
**Pay with PayPal**

**Pay Later**

**venmo**

**Pay by Card**

# 1.1.9. PayPal Review page



A Show PayPal mark, PayPal account and email address on order complete page

