

PayPal Create a transaction

Overview

Create a transaction. It can be a sale transaction or with a delayed capture.

Note: To choose between a sale transaction and a delayed capture, specify the field intent, which can be of the following types:

- CAPTURE - Directly capture funds when you create the transaction. No Authorization step is required before capture. You can capture up to the created transaction amount. When specified a higher amount, no error will be returned, instead the created amount value will be captured and a successful response will be returned. This is the default value for the intent field.
- AUTHORIZE - Create an **authorization**, which authorizes and places funds on hold for the order. Later, you can capture the funds for the order. For example, you can use the authorization and capture technique for orders containing items that are not immediately available for shipment. Then, you can create subsequent basic authorizations as the items become available. These authorizations ensure that the customer still has the funds available to purchase each item. By default, you can capture up to 115% of authorized amount or \$75, whichever is less. **This intent is not supported when you have more than one item within your transaction.**

Note: A valid JWT Token is required to make the API call.

Required fields

- amount - The `amount` object tells the total transaction amount
- amount.value - Amount to create. Always expects integer (112 will be 1.12 for USD/ EUR. For most currencies, 1000 is equal to 10; it depends on the number of digits after the decimal separator. You can see the list of all currencies supported by PayPal [here](#).
- amount.currencyCode - Currency code (ISO-4217)
- context - Transaction context
 - context.paymentProviderContract - Payment Provider contract ID from the Common Portal.

For Server-to-Server calls, since there is no front-end available, you can specify the `returnUrl` and the `cancelUrl` in the context object. These URLs would define where the customer would be redirected after the approval process is successful (`returnUrl`) or a failure (`cancelUrl`).

Note: These URLs are not mandatory to be passed in the request payload, but if you do not specify the `returnUrl` and the `cancelUrl`, then the customer would be stuck on PayPal portal only, if you are not using the PayPal JS SDK. With the SDK, these would not be used.

Optional fields

- intent - Intent of transaction. Defaults to CAPTURE.
- dynamicDescriptor - If an Order is paid using the "PayPal Wallet", the statement descriptor will appear in following format on the payer card statement: PAYPAL_prefix+(space)+merchant_descriptor+(space)+dynamicDescriptor. The PAYPAL prefix uses 8 characters. Only the first 22 characters will be displayed in the statement.
- merchantReference - The API caller-provided external ID. Used to reconcile client transactions with PayPal transactions. Appears in transaction and settlement reports but is not visible to the payer.
- description - Purchase description
- customer - Customer details object. Contains information about customer email address, name, phone number, address, etc.
- shipping - Shipping details object. Contains full name and address information.
- items - Items to purchase. Array of objects, that contains information about items category, sku, quantity, amount, etc.
- detailedAmount - Detailed amount object. Contains information about fees, discounts.
- context - Customize the payer experience during the approval process for the payment with PayPal.

Note: If you are providing **items** and **detailedAmount**, you must calculate **amount.value** with the following formula:

$$\text{itemTotal} = \text{item}[0].\text{unitAmount.value} * \text{item}[0].\text{quantity} + \text{item}[1].\text{unitAmount.value} * \text{item}[1].\text{quantity} + \dots + \text{item}[n].\text{unitAmount.value} * \text{item}[n].\text{quantity}$$

taxTotal = item[0].tax + item[1].tax + ... + item[n].tax

amount.value = itemTotal + taxTotal + detailedAmount.shipping.value + detailedAmount.handling.value + detailedAmount.insurance.value - detailedAmount.discount.value - detailedAmount.shippingDiscount.value

Here is an example POST request that can be used for creating a transaction.

```
POST /transactions
```

```
{
  "intent": "AUTHORIZE",
  "amount": {
    "value": 2700,
    "currencyCode": "USD"
  },
  "context": {
    "paymentProviderContract": "YOUR_PAYMENT_PROVIDER_CONTRACT"
  },
  "customer": {
    "email": "yourEmail@personal.example.com",
    "payerId": "WDJJHEBZ4X2LY",
    "phoneNumber": {
      "phoneType": "MOBILE",
      "value": "64646464"
    },
    "birthDate": "2000-01-31",
    "identification": {
      "taxIdentificationNumber": "123456",
      "taxIdentificationType": "BR_CNPJ"
    },
    "address": {
      "country": "IN",
      "postalCode": "570023",
      "countrySubdivision": "IN-MH",
      "city": "yyy",
      "addressLine1": "add1",
      "addressLine2": "add2"
    },
    "firstName": "James",
    "lastName": "Smith"
  },
  "applicationContext": {
    "brandName": "MAHENDRA",
    "locale": "he-IL",
    "landingPage": "BILLING",
    "returnUrl": "http://example.com/success",
    "cancelUrl": "http://example.com/failure"
  },
  "shipping": {
    "address": {
      "country": "IN",
      "postalCode": "91",
      "countrySubdivision": "IN-MH",
      "city": "mysore",
      "addressLine1": "walstreet",
      "addressLine2": "forcircle"
    },
    "fullName": "JamesSmith"
  }
}
```

```
"items": [
  {
    "name": "Mac Laptop",
    "unitAmount": {
      "currencyCode": "USD",
      "value": 100
    },
    "tax": {
      "currencyCode": "USD",
      "value": 100
    },
    "quantity": "1",
    "description": "Dell Laptop",
    "sku": "123",
    "category": "PHYSICAL_GOODS"
  },
  {
    "name": "Phone",
    "unitAmount": {
      "currencyCode": "USD",
      "value": 100
    },
    "tax": {
      "currencyCode": "USD",
      "value": 100
    },
    "quantity": "10",
    "description": "Apple phone",
    "sku": "456",
    "category": "PHYSICAL_GOODS"
  },
  {
    "name": "Dell XPS 9310 13.4 FHD Display Thin & Light 11th Gen Laptop (i5-1135G7 / 8 GB / 512 SSD / Integrated Graphics / 1Yr Premium",
    "unitAmount": {
      "currencyCode": "USD",
      "value": 100
    },
    "tax": {
      "currencyCode": "USD",
      "value": 100
    },
    "quantity": "1",
    "description": "groceries",
    "sku": "456",
    "category": "PHYSICAL_GOODS"
  },
  {
    "name": "Samsung 6.5 kg Fully-Automatic Top Loading Washing Machine (WA65A4002VS/TL, Imperial Silver, Center Jet Technology)",
    "unitAmount": {
      "currencyCode": "USD",
      "value": 100
    },
    "tax": {
      "currencyCode": "USD",
      "value": 100
    },
    "quantity": "1",
    "description": "product",
    "sku": "456",
    "category": "PHYSICAL_GOODS"
  }
],
"dynamicDescriptor": "Paypal order DD123",
"merchantReference": "DD123-reference",
"invoiceNumber": "INV12357",
```

```
"detailedAmount": {
  "discount": {
    "currencyCode": "USD",
    "value": 100
  },
  "shippingDiscount": {
    "currencyCode": "USD",
    "value": 100
  },
  "insurance": {
    "currencyCode": "USD",
    "value": 100
  },
  "handling": {
    "currencyCode": "USD",
    "value": 100
  },
  "shipping": {
    "currencyCode": "USD",
    "value": 100
  }
}
```

The response for create transaction will look like this:

```
{
  "id": "6a00b1be-3277-44b1-85bd-9f39a8e14b05",
  "orderId": "26W607012X848494T",
  "createdAt": "2021-03-26T10:36:25Z",
  "status": "INITIATED",
  "approvalUrl": "https://www.sandbox.paypal.com/checkoutnow?token=26W607012X841294T"
}
```

- id - The transaction ID that would be used in all the subsequent calls
- orderId - The corresponding PayPal order ID which is created for this transaction
- createdAt - The date and time when this transaction is created
- status - The current status of the transaction
- approvalUrl - The URL to redirect the customer to, for taking the approval for the current order on PayPal

The status of the transaction is INITIATED after this API call.

Get customer approval

When you create a Transaction for PayPal, the customer must approve the PayPal payment before you can authorize or capture the order. To get the approval from the customer, use the `approvalUrl` from the created transaction API response.

To test customer approval in sandbox:

1. Navigate to the `approvalUrl` given in the response of your create transaction call.
2. When the checkout flow popup appears, log in using the buyer account.
3. Make the payment.
4. Log into <https://www.sandbox.paypal.com/> with your buyer account. You will see the funds transfer from the buyer side.
5. Log into <https://www.sandbox.paypal.com/> with your facilitator account (also automatically created when you created a sandbox account) and you will see the funds transfer to you from the buyer.

When the customer approves the transaction on PayPal, the transaction status would change to APPROVED. Also, the detail of the customer who has approved the transaction would be available in the transaction details.