



PayPal 标准支付集成指南

- 用户按本指南操作后可使用 PayPal 标准支付收付款服务及相关功能, PayPal 标准支付收付款服务及相关功能属于 PayPal 全球服务 (定义如下)。用户需要首先在 PayPal 运营的中国网站 (Paypal.cn) 上接受 [PayPal 中国跨境支付服务用户协议](#) 并注册 PayPal 账户, 以进一步申请相关服务。PayPal 全球服务由我们合作的 PayPal 关联企业为您提供, 遵循并受到 “[PayPal 全球条款](#)” 的约束
- 本指南包含的全部产品交互示意图均为参考目的而展示, 用以阐释本指南之步骤引导。所有 PayPal 产品和服务均根据适用的 PayPal 用户协议、条款和政策按现状提供。

目录

1. 所需 SDK/API	4
2. 按钮效果展示	4
3. 标准支付按钮集成步骤 (图一):	6
步骤 1: 加载 PayPal JS 代码.....	6
步骤 2: 获取访问令牌	6
步骤 3: 创建订单	7
步骤 4: 返回订单 ID	8
步骤 5: 登录 PayPal 买家账号.....	8
步骤 6: 捕获支付	9
步骤 7: 处理支付结果	9
步骤 8: 处理异步通知 Webhooks.....	9
4. 信用卡支付按钮集成步骤 (图二):	10
步骤 1: 加载 JS 渲染信用卡按钮	10
步骤 2: 获取访问令牌	10
步骤 3: 创建订单	11
步骤 4: 返回订单 ID	12
步骤 5: 输入信用卡信息	13
步骤 6: 捕获支付	13
步骤 7: 处理支付结果	13
步骤 8: 处理异步通知: Webhooks.....	13

5. 参考文档.....	14
6. 附录.....	14
创建 sandbox 商家 Business 账号步骤.....	14
修改 Sandbox 密码步骤.....	15
创建 REST APP 步骤.....	16

1. 所需 SDK/API

集成此按钮所需的 SDK/API 如下:

- PayPal JS SDK: [PayPal JS SDK](#)
- Server SDK: [Server SDK](#) (Server SDK 目前支持的编程语言包括 PHP、Java、.Net、Python、Ruby 和 TypeScript)。
- API: [PayPal API](#)

2. 按钮效果展示

图一：标准支付按钮演示

Contact

Phone Number Email Address

Ship To

Address Line 1 Address Line 2


Nation and City


Shipping Method


Carrier


USPS Price: 20.00

Payment Method


 PayPal

 Debit or Credit Card

 G Pay

 Apple Pay

Item

 Test Refresh default Product

Amount: 1 Single Price: 100


Shipping 20

Total 120

[PayPal](#)

[Pay Later](#)

[Debit or Credit Card](#)

Powered by 

图二，单独信用卡支付按钮演示

Contact

Phone Number Email Address

Ship To

Address Line 1 Address Line 2


Nation and City


Shipping Method


Carrier


USPS Price: 20.00

Payment Method


 PayPal

 Debit or Credit Card

 G Pay

 Apple Pay

Item

 Test Refresh default Product

Amount: 1 Single Price: 100

Shipping 20

Total 120

[Debit or Credit Card](#)

✕

Card number

Expires CSC

Ship to billing address

By continuing, you confirm you're 18 years or older.

[Pay \\$120.00](#)

3. 标准支付按钮集成步骤（图一）：

步骤 1：加载 PayPal JS 代码

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

获取 client-id 的步骤请参见附录。JS 代码参考链接：[PayPal Checkout Demo](#)

```
// Render the PayPal button into #paypal-button-container
paypal.Buttons({
  // Call your server to set up the transaction
  createOrder: function(data, actions) {
    return fetch('/demo/checkout/api/paypal/order/create/', { // 服务器URL，调用PayPal Create Order API
      method: 'post'
    }).then(function(res) {
      return res.json();
    }).then(function(orderData) {
      return orderData.id; // return Create Order API 返回的order id
    });
  },

  // Call your server to finalize the transaction // 服务器URL，调用PayPal Capture API
  onApprove: function(data, actions) {
    return fetch('/demo/checkout/api/paypal/order/' + data.orderID + '/capture/', {
      method: 'post'
    }).then(function(res) {
      return res.json();
    }).then(function(orderData) {
      // Three cases to handle:
      // (1) Recoverable INSTRUMENT_DECLINED -> call actions.restart()
      // (2) Other non-recoverable errors -> Show a failure message
      // (3) Successful transaction -> Show confirmation or thank you

      // This example reads a v2/checkout/orders capture response, propagated from the server
      // You could use a different API or structure for your 'orderData'
      var errorDetail = Array.isArray(orderData.details) && orderData.details[0];

      if (errorDetail && errorDetail.issue === 'INSTRUMENT_DECLINED') {
        return actions.restart(); // Recoverable state, per:
        // https://developer.paypal.com/docs/checkout/integration-features/funding-failure/
      }

      if (errorDetail) {
        var msg = 'Sorry, your transaction could not be processed.';
        if (errorDetail.description) msg += '\n\n' + errorDetail.description;
        if (orderData.debug_id) msg += ' (' + orderData.debug_id + ')';
        return alert(msg); // Show a failure message (try to avoid alerts in production environments)
      }

      // Successful capture! For demo purposes:
      console.log('Capture result', orderData, JSON.stringify(orderData, null, 2));
      var transaction = orderData.purchase_units[0].payments.captures[0];
      alert('Transaction '+ transaction.status + ': ' + transaction.id + '\n\nSee console for all available details');

      // Replace the above to show a success message within this page, e.g.
      // const element = document.getElementById('paypal-button-container');
      // element.innerHTML = '';
      // element.innerHTML = '<h3>Thank you for your payment!</h3>';
      // Or go to another URL: actions.redirect('thank_you.html');
    });
  }
}).render('#paypal-button-container');
```

步骤 2：获取访问令牌

调用获取访问令牌 API：[获取访问令牌](#)

步骤 3: 创建订单

当点击 PayPal 智能按钮上的任何按钮时，会触发 JS 的 createOrder 函数，调用 Create Order 后台的 URL。在调用 Create Order 后台的 URL 时，调用 PayPal Create Order API。Create Order API 文档: [Create Order API](#)

必传项:

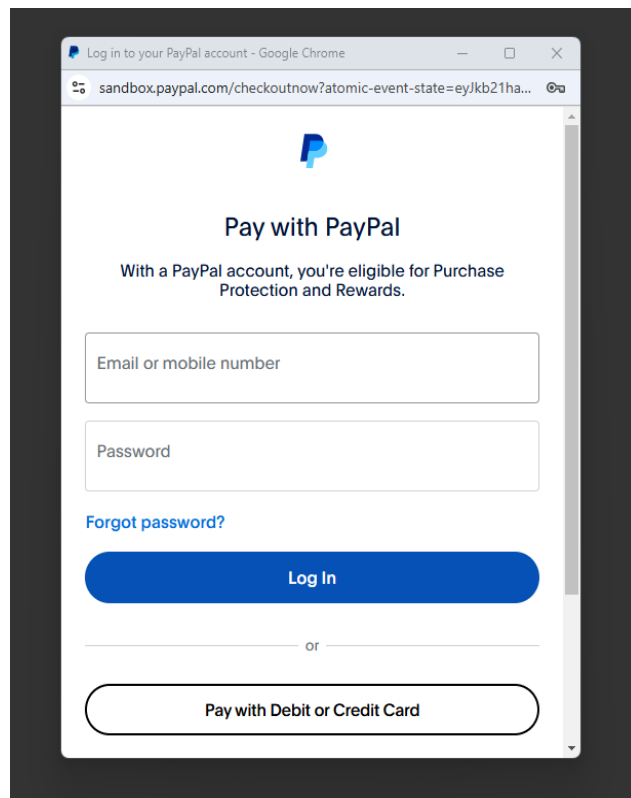
- 商品详情: 商品详情必填 purchase_units.items[]
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/items&t=request
- 实体商品:
 1. 商品种类选择实体商品 purchase_units.items[].name. category= PHYSICAL_GOODS
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/items/category&t=request
 2. 地址必传 purchase_units.shipping
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/shipping&t=request
 3. 如果买家在商家的网站上填写地址需要传 shipping_preference = SET_PROVIDED_ADDRESS
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=payment_source/paypal/experience_context/shipping_preference&t=request
- 非实体商品:
 1. 商品种类选择非实体商品 purchase_units.items[].name. category= DIGITAL_GOODS
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/items/category&t=request

2. 传 shipping_preference = NO_SHIPPING

https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=payment_source/paypal/experience_context/shipping_preference&t=request

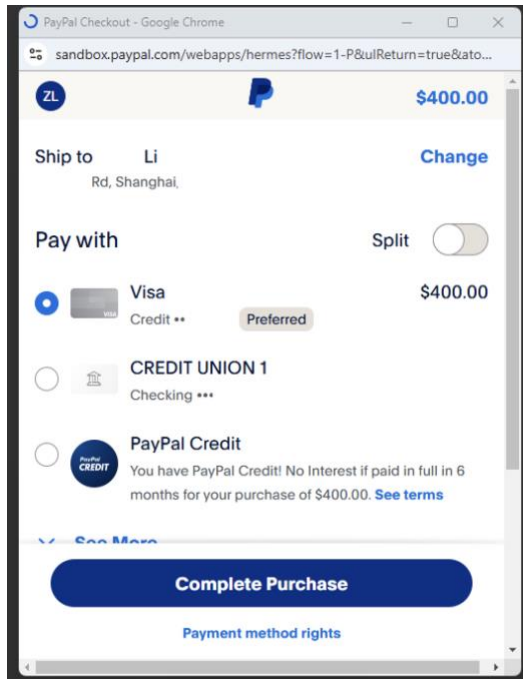
步骤 4: 返回订单 ID

JS createOrder 函数返回订单 ID。如果 API 调用成功，会弹窗并跳转到 PayPal 登录页面。



步骤 5: 登录 PayPal 买家账号

登录 PayPal 买家账号（创建 Sandbox 账号及修改密码的步骤请参见附录）后，点击“Complete Purchase”按钮时，会触发 JS onApprove 函数。



步骤 6: 捕获支付

在 JS onApprove 函数的 fetch()方法中传递订单 ID 请求服务器。服务器上调用 PayPal 的 Capture Payment API。 [Capture Payment API](#)

步骤 7: 处理支付结果

根据 Capture API 返回的信息判断交易是否成功，然后跳转到成功/失败页面（该页面为商家页面）。

步骤 8: 处理异步通知 Webhooks

如果 PayPal Capture API 首次返回的交易状态为 Pending，根据 Webhooks 来更新最终交易状态。Webhooks 文档: [Webhooks](#)

4. 信用卡支付按钮集成步骤 (图二):

步骤 1: 加载 JS 渲染信用卡按钮

加载 JS 代码以渲染信用卡按钮, 设置 `fundingSource: paypal.FUNDING.CARD` 单独显示信用卡按钮, 设置 `expandCardForm: true` 自动渲染信用卡表单。

```
<script>
var CARD_FUNDING_SOURCES = [
  paypal.FUNDING.CARD
];
// Loop over each funding source / payment method
CARD_FUNDING_SOURCES.forEach(function (fundingSource) {
  // Initialize the buttons
  var button = paypal.Buttons({
    fundingSource: fundingSource,
    expandCardForm: true,
    // Call your server to set up the transaction
    createOrder: function(data, actions) {
      return fetch('/demo/checkout/api/paypal/order/create/', {
        method: 'post'
      }).then(function(res) {
        return res.json();
      }).then(function(orderData) {
        return orderData.id;
      });
    },
    // Call your server to finalize the transaction
    onApprove: function(data, actions) {
      return fetch('/demo/checkout/api/paypal/order/' + data.orderID + '/capture/', {
        method: 'post'
      }).then(function(res) {
        return res.json();
      }).then(function(orderData) {
        var errorDetail = Array.isArray(orderData.details) && orderData.details[0];

        if (errorDetail && errorDetail.issue === 'INSTRUMENT_DECLINED') {
          return actions.restart(); // Recoverable state, per:
          // https://developer.paypal.com/docs/checkout/integration-features/funding-failure/
        }

        if (errorDetail) {
          var msg = 'Sorry, your transaction could not be processed.';
          if (errorDetail.description) msg += '\n\n' + errorDetail.description;
          if (orderData.debug_id) msg += ' (' + orderData.debug_id + ')';
          return alert(msg); // Show a failure message (try to avoid alerts in production environments)
        }

        // Successful capture! For demo purposes:
        console.log('Capture result', orderData, JSON.stringify(orderData, null, 2));
        var transaction = orderData.purchase_units[0].payments.captures[0];
        alert('Transaction ' + transaction.status + ': ' + transaction.id + '\n\nSee console for all available details');
      });
    }
  });

  // Check if the button is eligible
  if (button.isEligible()) {
    // Render the standalone button for that funding source
    button.render('#paypal-button-container');
  }
});
</script>
```

步骤 2: 获取访问令牌

调用获取访问令牌 API: [获取访问令牌](#)

步骤 3: 创建订单

当点击黑色信用卡按钮时, 会触发 JS 的 createOrder 函数, 调用 Create Order 后台的 URL。在调用 Create Order 后台的 URL 时, 调用 PayPal Create Order API。Create Order API 文档: [Create Order API](#)。调用 Create Order API 时需要传递 Payer 信息, 这样在信用卡下拉框显示时会隐藏这些信息。

```
payer: {
  name: {
    given_name: "PayPal",
    surname: "Customer"
  },
  address: {
    address_line_1: 'オオイタケン, ナカツシ, ヤバケイマチカブシノ, 213-1173',
    address_line_2: '213-1173',
    admin_area_2: 'Oita',
    admin_area_1: 'Oita',
    postal_code: '871-0402',
    country_code: 'JP'
  },
  email_address: "customer@domain.com",
  phone: {
    phone_number: {
      national_number: "555555555"
    }
  }
},
```

必传项:

- 商品详情: 商品详情必填 purchase_units.items[]
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/items&t=request
- 实体商品:
 1. 商品种类选择实体商品 purchase_units.items[].name.category=PHYSICAL_GOODS
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/items/category&t=request
 2. 地址必传 purchase_units.shipping
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/shipping&t=request

3. 如果买家在商家的网站上填写地址需要传 shipping_preference = SET_PROVIDED_ADDRESS
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=payment_source/paypal/experience_context/shipping_preference&t=request
- 非实体商品:
 3. 商品种类选择非实体商品 purchase_units.items[].name.
category= DIGITAL_GOODS
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/items/category&t=request
 4. 传 shipping_preference = NO_SHIPPING
https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=payment_source/paypal/experience_context/shipping_preference&t=request

步骤 4: 返回订单 ID

JS createOrder 函数返回订单 ID。如果 API 调用成功，信用卡表单将以下拉框的形式展现。

The image shows a PayPal payment interface. At the top, there are radio buttons for selecting a payment method: PayPal, MasterCard, VISA, DISCOVER, and AMERICAN EXPRESS. The MasterCard option is selected. Below this is a dark grey button labeled "Debit or Credit Card" with a close icon (X) to its right. Underneath are three input fields: "Card number", "Expires", and "CSC". Below the input fields is a checkbox with the text "I confirm that I am of legal age and agree to the PayPal [Privacy Statement](#)." At the bottom is a large blue button labeled "Pay \$200.00".

步骤 5：输入信用卡信息

买家输入信用卡信息（Sandbox 上的测试信用卡卡号可以在以下链接生成：[信用卡生成器](#)）后，点击“Pay”按钮会触发 JS onApprove 函数。

步骤 6：捕获支付

在 JS onApprove 函数的 fetch()方法中传递订单 ID 请求服务器。服务器上调用 PayPal 的 Capture Payment API。[Capture Payment API](#)

步骤 7：处理支付结果

根据 Capture API 返回的信息判断交易是否成功，然后跳转到成功/失败页面（该页面为商家页面）。

步骤 8：处理异步通知：Webhooks

如果 PayPal Capture API 首次返回的交易状态为 Pending，根据 Webhooks 来更新最终交易状态。Webhooks 文档：[Webhooks](#)

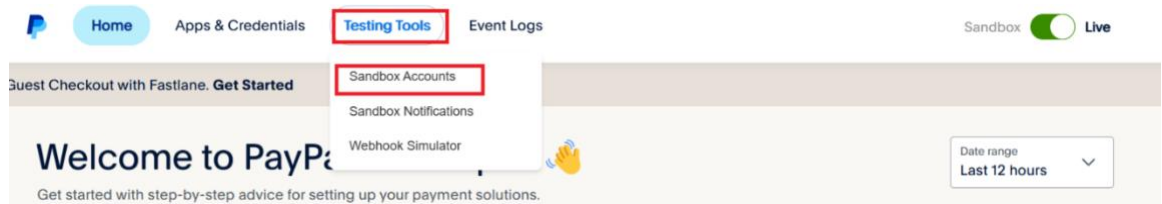
5. 参考文档

- 标准支付文档: [PayPal 标准支付文档](#)
- API 文档: [PayPal API 文档](#)
- Server SDK: [PayPal Server SDK](#)

6. 附录

创建 sandbox 商家 Business 账号步骤

1. 访问 [PayPal 开发者主页](#)
2. 点击右上角的 "Log In"
3. 使用真实的 PayPal 收款账号登录
4. 点击 "Testing Tools" 下的 "Sandbox Accounts"



5. 点击 "Create account"
6. 选择 "Business", 选择 "China" 后点击 "Create Account" (如果需要创建 Sandbox 买家账号, 在此步骤选择 "Personal", 国家选择非 China。)

Create Sandbox Account

Choose your account type and country that will be used for live payments.

Account type:

- Personal**
Buyer account
- Business**
Merchant account

Country/Region:


Country or region
China

Create Account

Do you want a more customized account? [Create](#)

修改 Sandbox 密码步骤

1. 访问 [PayPal 开发者主页](#)
2. 点击右上角的 "Log In"
3. 使用真实的 PayPal 收款账号登录
4. 点击 "Testing Tools" 下的 "Sandbox Accounts"
5. 找出需要更换密码的 Sandbox 账号，点击最右边的 "View/Edit"

<input type="checkbox"/>	sb-nq4oe15362897@business.example.com	Business	C2	11/20/24, 9:09 AM	
<input type="checkbox"/>	sb-bv43xx33885946@personal.example.com	Personal	US	11/8/24	View/Edit account

6. 点击 "Change Password" 修改密码

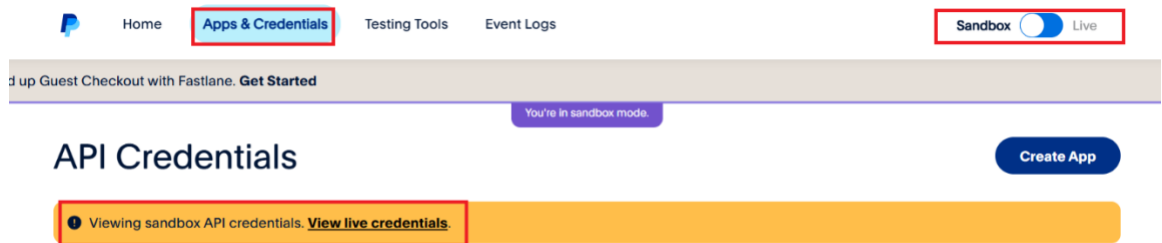
[sb-nq4oe15362897@business.example.com](#)

Login Info

Sandbox URL	https://sandbox.paypal.com
Email	sb-nq4oe15362897@business.example.com Change email
Password Change password

创建 REST APP 步骤

1. 访问 [PayPal 开发者主页](#)
2. 点击右上角的 "Log In"
3. 使用真实的 PayPal 收款账号登录
4. 点击 "App & Credentials" (注: 右上角可以切换 Sandbox/Live 环境下创建)



5. 点击 "Create App"
6. 输入 App 名称 (只要不重复, 可以随意命名), 选择上面刚刚创建的 Sandbox Business 账号

A screenshot of the 'Create App' form. It features a text input field for 'App Name' with the value 'Standard'. Below this is a 'Type:' section with two radio button options: 'Merchant' (selected) and 'Platform'. The 'Merchant' option is described as 'Accept payments as a merchant (seller)'. The 'Platform' option is described as 'Move payments to sellers as a platform (marketplace, crowdfunding, or e-commerce platform)'. Below the type selection is a 'Sandbox Account:' section with a dropdown menu showing 'Sandbox Business Account' and the email address 'sb-nq4oe15362897@business.example.com (...)' (the dropdown arrow is highlighted with a red box). At the bottom, there is a disclaimer: 'By clicking the button below, you agree to PayPal Developer Agreement (US accounts only).' and a large blue 'Create App' button.

7. 点击 "Create App", 获取 clientid 和 Secret.