



PayPal 保存付款集成指南

- 用户按本指南操作后可使用 PayPal 保存付款服务及相关功能, PayPal 保存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.....	5
3. 保存付款方式按钮集成步骤: .....	7
3.1. 在购物期间保存付款方式按钮集成步骤: .....	7
步骤 1: 获取 user ID token.....	7
步骤 2: 加载 PayPal JS 代码.....	7
步骤 3: 创建订单.....	8
步骤 4: 返回订单 ID .....	10
步骤 5: 登录 PayPal 买家账号 .....	10
步骤 6: 捕获支付.....	11
步骤 7: 处理支付结果 .....	11
步骤 8: 处理异步通知 Webhooks.....	12
3.2. 在购物前保存付款方式按钮集成步骤: .....	12
步骤 1: 获取 user ID token.....	12
步骤 2: 加载 PayPal JS 代码.....	12
步骤 3: 创建 Setup Token.....	13
步骤 4: 创建 Payment Token .....	13
步骤 5: 登录 PayPal 买家账号 .....	14
步骤 6: 捕获 Vault setup token ID .....	15
步骤 7: 处理付款令牌数据 .....	15
步骤 8: 处理异步通知 Webhooks.....	15
4. 已保存支付用户: .....	16
4.1 由买家发起的已保存支付用户的扣款 .....	16

步骤 1: 获取 user ID token.....	16
步骤 2: 加载 PayPal JS 代码.....	16
步骤 3: 创建订单.....	17
步骤 4: 返回订单 ID .....	19
步骤 5: 捕获支付.....	19
步骤 6: 处理支付结果 .....	19
步骤 7: 处理异步通知 Webhooks.....	19
4.2 由商家发起的已保存支付用户的扣款 .....	19
步骤 1: 获取访问令牌 .....	19
步骤 2: 创建订单 .....	20
步骤 3: 处理支付结果 .....	21
步骤 4: 处理异步通知: Webhooks .....	21
5. 参考文档 .....	21
6. 附录 .....	21
创建 sandbox 商家 Business 账号步骤 .....	21
修改 Sandbox 密码步骤.....	22
创建 REST APP 步骤.....	23
Sandbox 账户开通 vault 权限 .....	24
Live 账户开签约支付权限.....	25

## 1. 所需 SDK/API

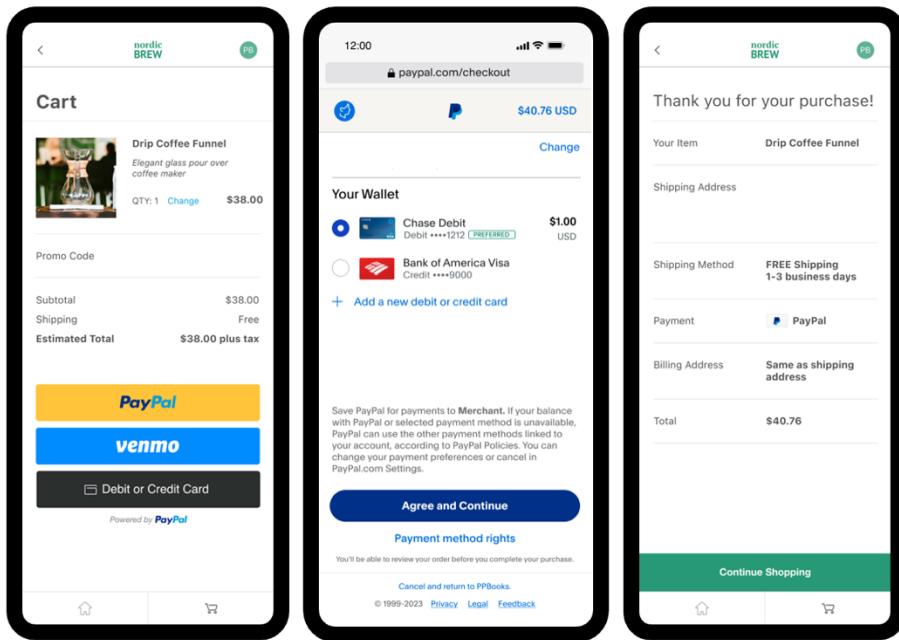
集成此按钮所需的 SDK/API 如下:

- PayPal JS SDK: [PayPal JS SDK](#)

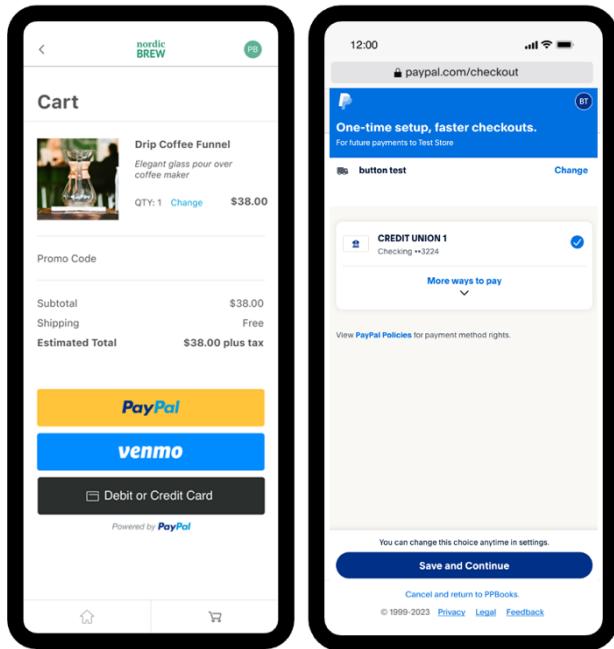
- Server SDK: [Server SDK](#) (Server SDK 目前支持的编程语言包括 PHP、Java、.Net、Python、Ruby 和 TypeScript)。
- API: [PayPal API](#)

## 2. 按钮效果展示

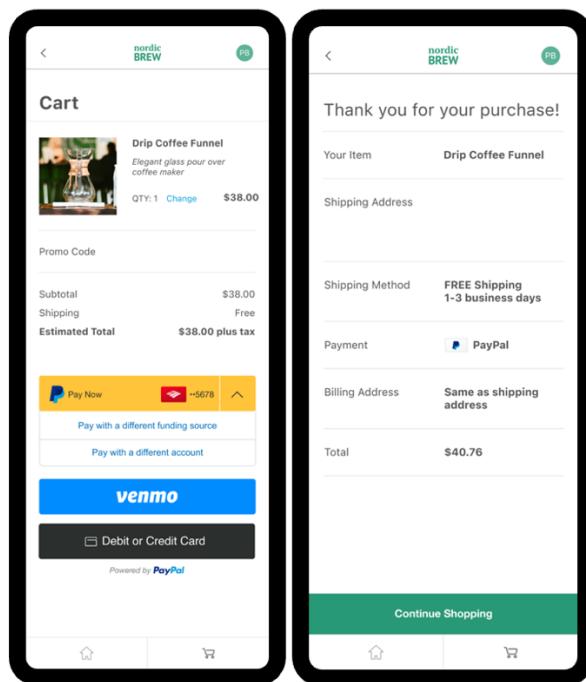
在购物期间保存付款方式流程演示：



在购物前保存付款方式流程演示：



## 保存支付用户返回付款流程演示：



### 3. 保存付款方式按钮集成步骤：

#### 3.1. 在购物期间保存付款方式按钮集成步骤：

##### 步骤 1：获取 user ID token

```
1 curl -s -X POST "https://api-m.sandbox.paypal.com/v1/oauth2/token" \
2   -u CLIENT_ID:CLIENT_SECRET \
3   -H "Content-Type: application/x-www-form-urlencoded" \
4   -d "grant_type=client_credentials" \
5   -d "response_type=id_token"
```

返回值中会包含 access\_token, id\_token, 以及 access\_token 可用时间

##### 步骤 2：加载 PayPal JS 代码

```
<script src="https://www.paypal.com/sdk/js?client-id=YOUR_CLIENT_ID"
data-user-id-token="YOUR-ID-TOKEN"></script>
```

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

```

1 <script>
2   paypal.Buttons({
3     // Call your server to set up the transaction
4     createOrder: function(data, actions) {    服务器 URL, 调用 PayPal Create Order API
5       return fetch('/yourserver.com/createOrder', {
6         method: 'post',
7         body: JSON.stringify({
8           source: data.paymentSource, //paypal / venmo / etc.
9         },
10        // Here for product info
11      }).then(function(res) {
12        return res.json();
13      }).then(function(orderData) {
14        return orderData.id;          return Create Order API 返回的 order id
15      });
16    },
17    // Authorize or capture the transaction after payer approves
18    onApprove: (data, actions) => {
19      return fetch('/yourserver.com/order/' + data.orderID + '/capture/', {
20        method: 'post'                  服务器 URL, 调用 PayPal Capture Order API
21      });
22    },
23    onCancel(data, actions) {
24      console.log(`Order Canceled - ID: ${data.orderID}`);
25    },
26    onError(err) {
27      console.error(err);
28    }
29  }).render('#paypal-button-container');
30 </script>

```

## 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：

- 签约交易必传项 <https://developer.paypal.com/docs/checkout/save-payment-methods/during-purchase/js-sdk/paypal/#link-request>：
  - `payment_source.paypal.attributes.vault.store_in_vault = ON_SUCCESS`

[https://developer.paypal.com/docs/api/orders/v2/#orders\\_create!path=payment\\_source/paypal/attributes/vault&t=request](https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=payment_source/paypal/attributes/vault&t=request)

2. `payment_source.paypal.experience_context.return_url`  
`payment_source.paypal.experience_context.r.cancel_url`  
[https://developer.paypal.com/docs/api/orders/v2/#orders\\_create!path=payment\\_source/paypal/experience\\_context&t=request](https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=payment_source/paypal/experience_context&t=request)

- 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](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](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](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](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`

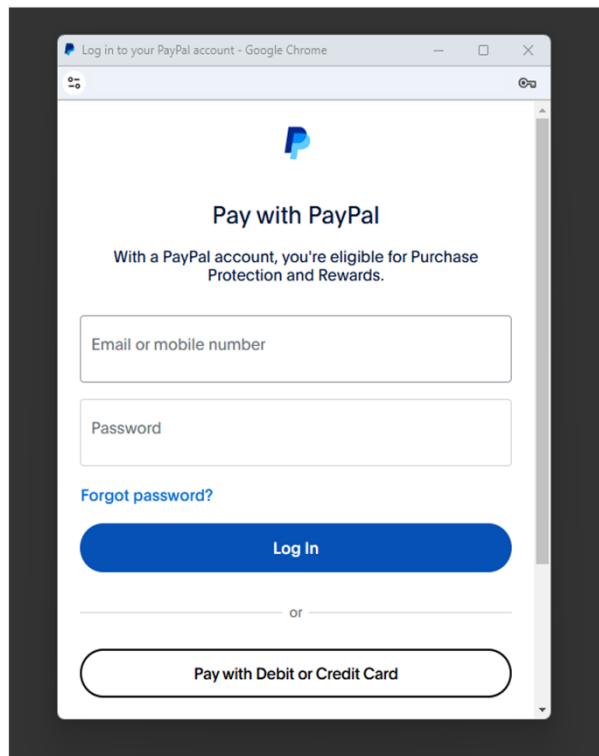
[https://developer.paypal.com/docs/api/orders/v2/#orders\\_create!path=purchase\\_units/items/category&t=request](https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/items/category&t=request)

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

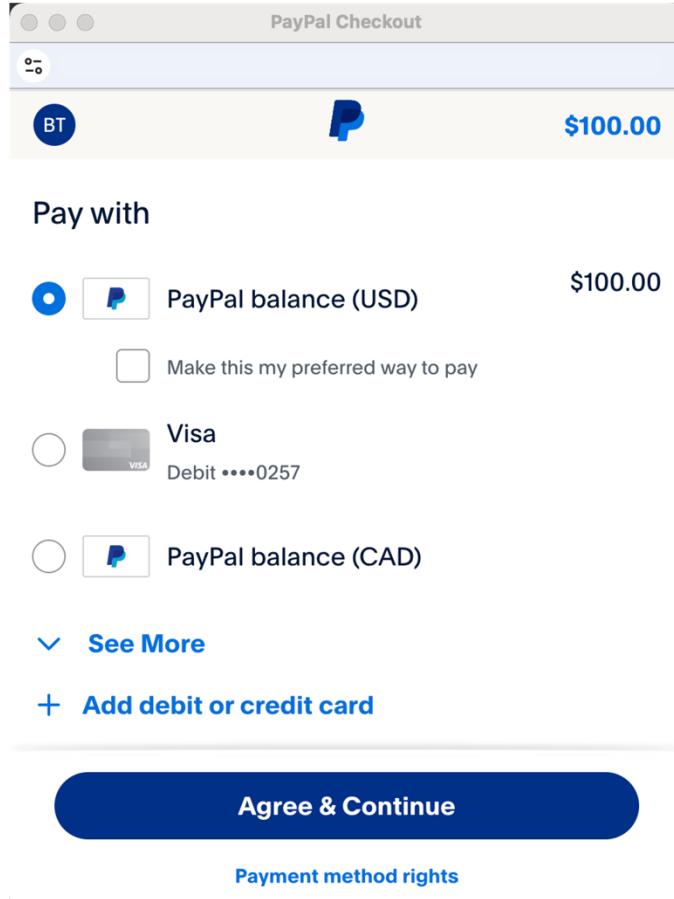
## 步骤 4：返回订单 ID

JS 的 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“Agree&Continue”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）。<https://developer.paypal.com/docs/checkout/save-payment-methods/during-purchase/js-sdk/paypal/#link-captureorderresponse>

查看 `vault.status`, `purchase_units`.`Payments`.`Captures`.`status` 记录返回值中的 `vault.id`, `vault.customer.id`, `purchase_units`.`Payments`.`Captures`.`id`,

## 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](#)

## 3.2. 在购物前保存付款方式按钮集成步骤：

### 步骤 1：获取 user ID token

```
1 curl -s -X POST "https://api-m.sandbox.paypal.com/v1/oauth2/token" \
2   -u CLIENT_ID:CLIENT_SECRET \
3   -H "Content-Type: application/x-www-form-urlencoded" \
4   -d "grant_type=client_credentials" \
5   -d "response_type=id_token"
```

返回值中会包含 access\_token, id\_token, 以及 access\_token 可用时间

### 步骤 2：加载 PayPal JS 代码

```
<script src="https://www.paypal.com/sdk/js?client-id=YOUR_CLIENT_ID"
data-user-id-token="YOUR-ID-TOKEN"></script>
```

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

```

<script>
  window.paypal.Buttons({
    createVaultSetupToken: async () => {
      // Call your server API to generate a setup token
      // and return it here as a string 服务器 URL, 调用 PayPal Create Vault Setup Token API
      const result = await fetch("example.com/create/setup/token", { method: "POST" })
      return result.token // return Create Vault Setup Token 返回的 id
    },
    onApprove: async ({ vaultSetupToken }) => {
      return fetch("example.com/create/payment/token", { body: JSON.stringify({ vaultSetupToken }) })
    },
    onError: (error) => {
      console.log("An error occurred: ", error)
    }
  }).render("#paypal-buttons-container");
</script>

```

## 步骤 3：创建 Setup Token

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

必传项：

签约交易必传项：

1. `payment_source.paypal.usage_type=MERCHANT`

[https://developer.paypal.com/docs/api/payment-tokens/v3/#setup-tokens\\_create!ct=application/json&path=payment\\_source&t=request](https://developer.paypal.com/docs/api/payment-tokens/v3/#setup-tokens_create!ct=application/json&path=payment_source&t=request)

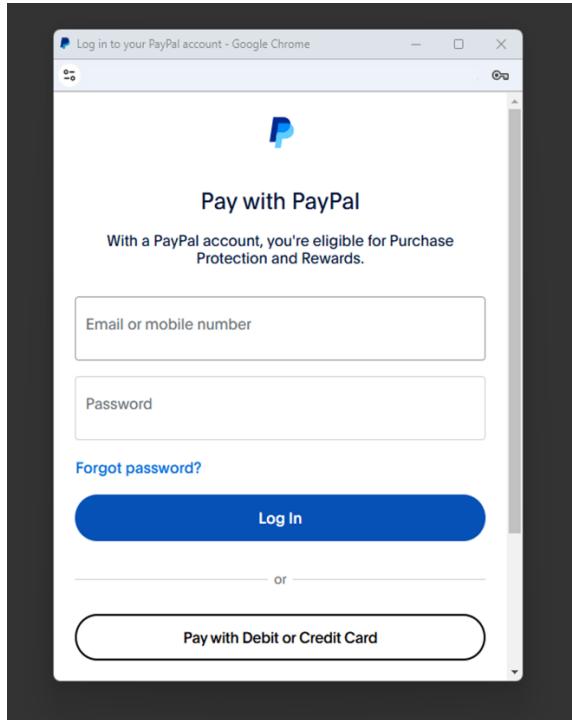
2. `payment_source.paypal.experience_context.return_url`

`payment_source.paypal.experience_context.r.cancel_url`

[https://developer.paypal.com/docs/api/payment-tokens/v3/#setup-tokens\\_create!ct=application/json&path=payment\\_source/paypal/experience\\_context&t=request](https://developer.paypal.com/docs/api/payment-tokens/v3/#setup-tokens_create!ct=application/json&path=payment_source/paypal/experience_context&t=request)

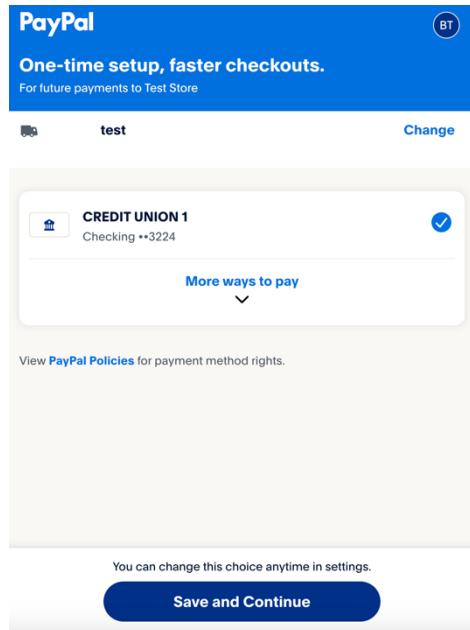
## 步骤 4：创建 Payment Token

JS `createVaultSetupToken` 函数返回 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“Save and Continue”按钮时，会触发 JS onApprove 函数。



## 步骤 6：捕获 Vault setup token ID

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 Create payment token API。[Create payment token API](#)

必传项：payment\_source.token.type= SETUP\_TOKEN, payment\_source.token.id

<https://developer.paypal.com/docs/api/payment-tokens/v3/#payment-tokens-create>

## 步骤 7：处理付款令牌数据

根据 Create payment token API 返回的信息判断授权是否成功，然后跳转到成功/失败页面（该页面为商家页面）。

<https://developer.paypal.com/docs/checkout/save-payment-methods/purchase-later/js-sdk/paypal/#link-sampleapiresponse>

查看记录返回值中的 id, customer.id

## 步骤 8：处理异步通知 Webhooks

如果第七部返回值因网络原因丢失，根据 Webhooks 来更新最终 payment token 状态。Webhooks 文档：[Webhooks](#)

## 4. 已保存支付用户：

### 4.1 由买家发起的已保存支付用户的扣款

#### 步骤 1：获取 user ID token

```
1 curl -s -X POST "https://api-m.sandbox.paypal.com/v1/oauth2/token" \
2   -u CLIENT_ID:CLIENT_SECRET \
3   -H "Content-Type: application/x-www-form-urlencoded" \
4   -d "grant_type=client_credentials" \
5   -d "response_type=id_token" \
6   -d "target_customer_id=CUSTOMER-ID"
```

API 中需要提供 target\_customer\_id，该 ID 在初次保存支付时来源于 payment\_source 中的 vault.customer.id。

返回值中会包含 access\_token, id\_token, 以及 access\_token 可用时间

#### 步骤 2：加载 PayPal JS 代码

```
<script src="https://www.paypal.com/sdk/js?client-id=YOUR_CLIENT_ID"
data-user-id-token="YOUR-ID-TOKEN"></script>
```

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

```

1 <script>
2   paypal.Buttons({
3     // Call your server to set up the transaction
4     createOrder: function(data, actions) {    服务器 URL, 调用 PayPal Create Order API
5       return fetch('/yourserver.com/createOrder', {
6         method: 'post',
7         body: JSON.stringify({
8           source: data.paymentSource, //paypal / venmo / etc.
9         }),
10        // Here for product info
11      }).then(function(res) {
12        return res.json();
13      }).then(function(orderData) {
14        return orderData.id;          return Create Order API 返回的 order id
15      });
16    },
17    // Authorize or capture the transaction after payer approves
18    onApprove: (data, actions) => {
19      return fetch('/yourserver.com/order/' + data.orderID + '/capture/', {
20        method: 'post'
21      });                      服务器 URL, 调用 PayPal Capture Order API
22    },
23    onCancel(data, actions) {
24      console.log(`Order Canceled - ID: ${data.orderID}`);
25    },
26    onError(err) {
27      console.error(err);
28    }
29  }).render('#paypal-button-container');
30 </script>

```

## 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：

- 签约交易后续支付必传项：
  - <https://developer.paypal.com/docs/checkout/save-payment-methods/during-purchase/js-sdk/paypal/#link-request>：
   
`payment_source.paypal.experience_context.return_url`
  
`payment_source.paypal.experience_context.r.cancel_url`

[https://developer.paypal.com/docs/api/orders/v2/#orders\\_create!path=payment\\_source/paypal/experience\\_context&t=request](https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=payment_source/paypal/experience_context&t=request)

2. 实体商品必须传参数: payment\_source.paypal.experience\_context.shipping\_preference=SET\_PROVIDED\_ADDRESS

虚拟商品必须传参数: payment\_source.paypal.experience\_context.shipping\_preference=NO\_SHIPPING

参考:

[https://developer.paypal.com/docs/api/orders/v2/#orders\\_create!ct=application/json&path=payment\\_source/paypal/experience\\_context/shipping\\_preference&t=request](https://developer.paypal.com/docs/api/orders/v2/#orders_create!ct=application/json&path=payment_source/paypal/experience_context/shipping_preference&t=request)

- 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](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](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](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](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](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](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，会触发 JS onApprove 函数。

## 步骤 5：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 6：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）。<https://developer.paypal.com/docs/checkout/save-payment-methods/during-purchase/js-sdk/paypal/#link-captureorderresponse>

查看 purchase\_units. Payments. Captures.status 记录返回值中的 purchase\_units. Payments. Captures.id

## 步骤 7：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](#)

## 4.2 由商家发起的已保存支付用户的扣款

### 步骤 1：获取访问令牌

调用获取访问令牌 API：[获取访问令牌](#)

## 步骤 2：创建订单

调用 PayPal Create Order API。Create Order API 文档：[Create Order API](#)。调用 Create Order API 时需要传递 vault\_id 信息，

必传项：

- 在 HTTP request headers 中确保 PayPal-Request-Id 的传参  
<https://developer.paypal.com/api/rest/requests/#link-httpprerequestheaders>
- payment\_source.paypal.vault\_id， 来自初次签约 capture order 后的返回值 <https://developer.paypal.com/docs/checkout/save-payment-methods/purchase-later/payment-tokens-api/paypal/#link-usesavedpaymenttoken>
- 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](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](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](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\\_preferece&t=request](https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=payment_source/paypal/experience_context/shipping_preferece&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](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](https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=payment_source/paypal/experience_context/shipping_preference&t=request)

## 步骤 3：处理支付结果

根据步骤 2 的 Create Order 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 4：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

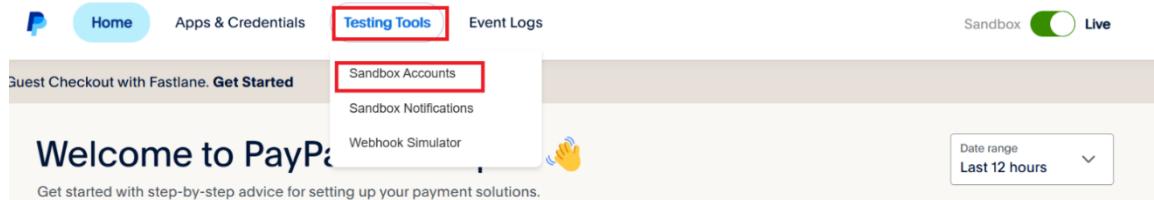
- 签约支付文档: [PayPal 签约支付文档](#)
- API 文档: [PayPal API 文档](#)
- Server SDK: [PayPal Server SDK](#)

## 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="button" value="..."/>
<input type="checkbox"/> sb-bv43xx33885946@personal.example.com	Personal	US	11/8/24	<input type="button" value="View/Edit account"/>

## 6. 点击 "Change Password" 修改密码

sb-nq4oe15362897@business.example.com

### Login Info

Sandbox URL	<a href="https://sandbox.paypal.com">https://sandbox.paypal.com</a>
Email	sb-nq4oe15362897@business.example.com <input type="button" value="Change email"/>
Password	..... <input type="button" value="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)

The screenshot shows the PayPal developer portal interface. At the top, there's a navigation bar with 'Home', 'Apps & Credentials' (which is highlighted with a red box), 'Testing Tools', and 'Event Logs'. To the right of the navigation is a toggle switch for 'Sandbox' (set to 'Live'). Below the navigation, a banner says 'Get Started' and 'You're in sandbox mode.' A red box highlights a message: 'Viewing sandbox API credentials. [View live credentials.](#)' On the right side of the main area, there's a 'Create App' button.

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

App Name  
Standard

Type:

**Merchant**  
Accept payments as a merchant (seller)

**Platform**  
Move payments to sellers as a platform (marketplace, crowdfunding, or e-commerce platform)

Sandbox Account:

Sandbox Business Account  
sb-nq4oe15362897@business.example.com (... ▾)

By clicking the button below, you agree to [PayPal Developer Agreement](#) (US accounts only).

**Create App**

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

## Sandbox 账户开通 vault 权限

1. 点击刚刚创建 Rest App Name
2. 找到 Features 设置确认 vault 是否已经选中

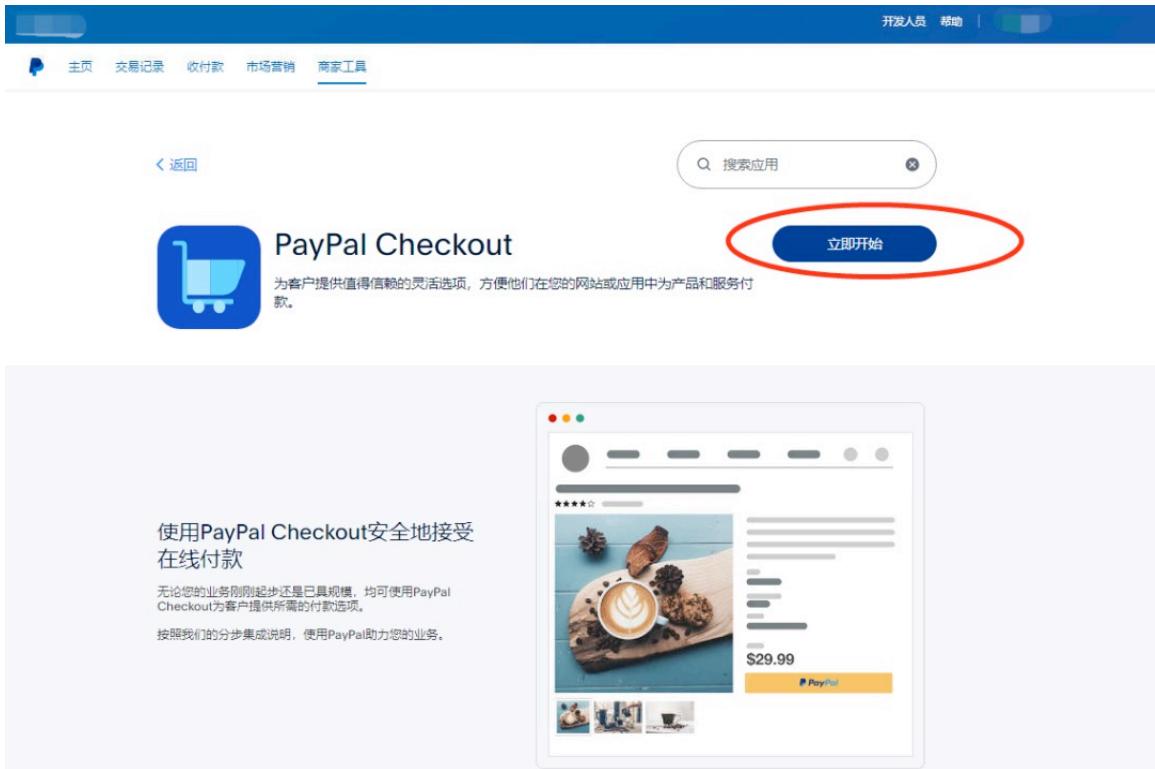
The screenshot shows the PayPal Developer Dashboard. At the top, there's a navigation bar with links for Docs, APIs & SDKs, Tools, Video Library, Help, and Business Dashboard. Below the navigation is a menu bar with Home, Apps & Credentials, Testing Tools, Event Logs, and a Sandbox link. A red box highlights the 'Features' button in the main content area. The content area has two sections: 'Accept payments' and 'Other features'. Under 'Accept payments', there are several options: Advanced Credit and Debit Card Payments (checked), Apple Pay (unchecked), Google Pay (unchecked), IC++ with Gross Settlement (unchecked), and Vault (checked). A red box highlights the 'Vault' section. Under 'Other features', there are four options: Payouts (checked), Customer disputes (checked), Log in with PayPal (unchecked), and Transaction search (unchecked).

## Live 账户开签约支付权限

1. 使用真实 PayPal 账号登录 [PayPal 主页](#)
2. 点击“商家工具”
3. 点击“PayPal Checkout”

The screenshot shows the PayPal Merchant Tools page. At the top, there's a navigation bar with links for 主页, 交易记录, 收付款, 市场营销, and 商家工具 (which is circled in red). Below the navigation is a search bar with a placeholder '搜索应用' and a clear button. The main content area is titled '商家工具' and contains two options: '接受付款' (PayPal Checkout) and '循环付款管理平台' (Circular Payment Management Platform). The 'PayPal Checkout' option is highlighted with a red oval.

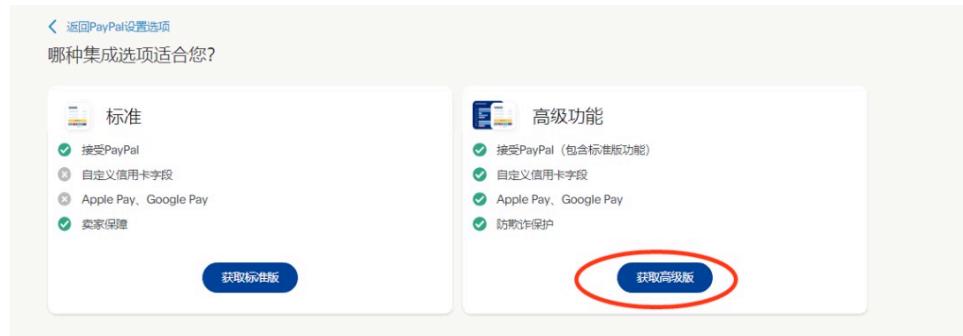
4. 点击“立即开始”



## 5. 点击有定制网站下面的“选择”



## 6. 点击“获取高级版”



7. 输入信息后，“同意并提交”

8. 已获准使用签约支付功能



祝贺您，！您已获准使用PayPal Complete Payments功能

只要您准备就绪，就可在您的网站上开始设置PayPal。

[前往首页](#)

[在您的网站上设置PayPal](#)