PayPal 高级版支付集成指南

- 用户按本指南操作后可使用 PayPal 高级版支付收付款服务及相关功能, PayPal 高级版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

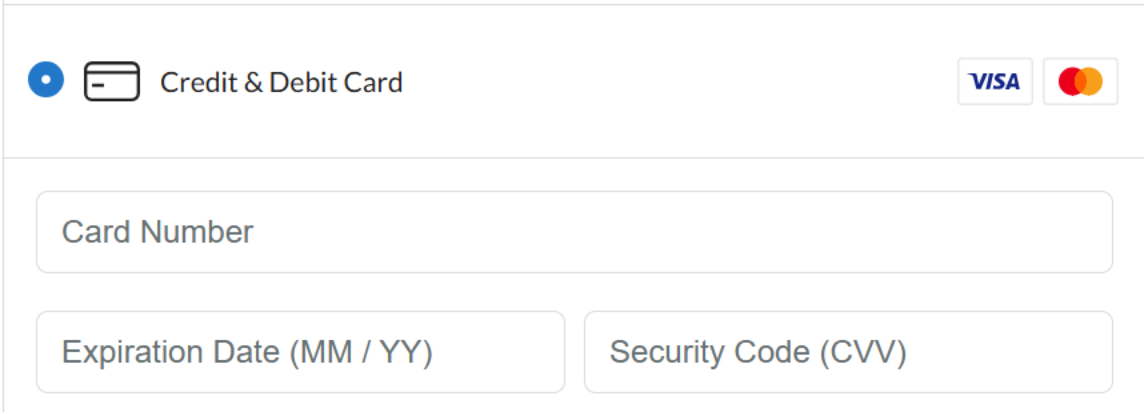
1. 所需 SDK/API

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

高级信用卡支付功能展示

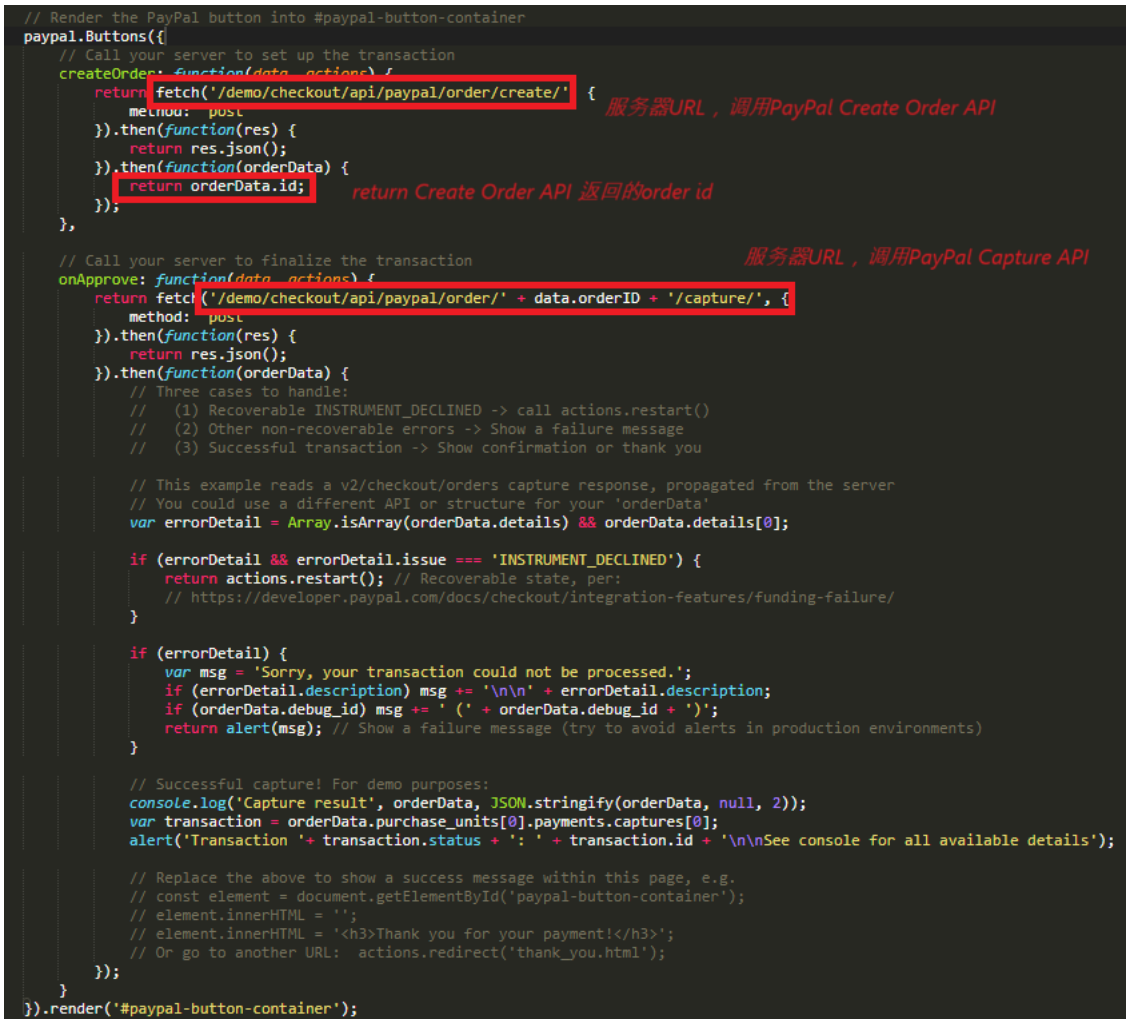


3. 标准支付按钮集成步骤：

步骤 1：加载 PayPal JS 代码

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



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

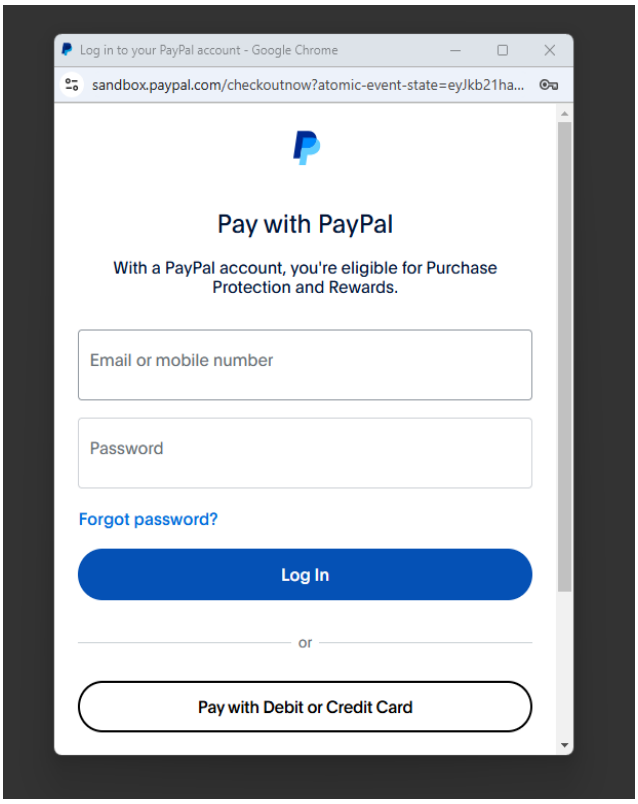
必传项：

- 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

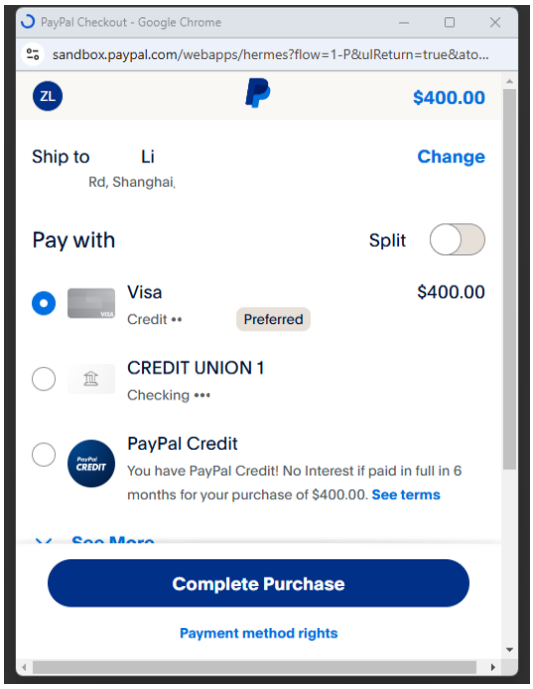
步骤 4：返回订单 ID

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. 高级信用卡支付按钮集成步骤

步骤 1：加载 JS

<script src="https://www.paypal.com/sdk/js?components=card-fields&disable-funding=card&client-id=<your_client_id>"></script> 获取 client-id 的步骤请参见附录。

注： 如需启用高级信用卡功能，则需要关掉标准信用卡功能，添加参数&disable-funding=card

步骤 2：通过 JS 加载按钮

在加载按钮前，确保您拥有高级信用卡的权限。查看权限步骤请参见附录。详细文档请参考：高级信用卡支付集成文档

步骤 3：创建高级信用卡支付表单

创建高级信用卡支付表单。PayPal 的 JS SDK 提供了一个组件 cardFields，该组件用于接受和保存卡片信息。使用该组件加载信用卡支付表单，PayPal 会处理卡片相关的所有安全和合规问题。通过将每个卡片字段声明为 cardField 中的一个对象，然后应用.render()函数来渲染每个卡片字段。

```
if (cardField.isEligible()) {
    const nameField = cardField.NameField({
        style: { input: { color: "blue" }, ".invalid": { color: "purple" } },
    });
    nameField.render("#card-name-field-container");

    const numberField = cardField.NumberField({
        style: { input: { color: "blue" } },
    });
    numberField.render("#card-number-field-container");

    const cvvField = cardField.CVVField({
        style: { input: { color: "blue" } },
    });
    cvvField.render("#card-cvv-field-container");

    const expiryField = cardField.ExpiryField({
        style: { input: { color: "blue" } },
    });
    expiryField.render("#card-expiry-field-container");
```

表单的样式可以通过 CSS 编辑，详细文档请参考：自定义信用卡字段样式；CSS 案例：CSS 案例

```
const cardField = window.paypal.CardFields({
    createOrder: createOrderCallback,
    onApprove: onApproveCallback,
    style: {
        input: {
            "font-size": "16px",
            "font-family": "courier, monospace",
            "font-weight": "lighter",
            color: "#ccc",
        },
        ".invalid": { color: "purple" },
    },
});
```

步骤 4：买家输入信用卡信息

买家在高级信用卡表单上输入信用卡、CVV、有效期信息后（sandbox 环境下的测试信用卡信息可以在以下链接获取：测试信用卡信息），点击支付按钮（该按钮为自定义按钮，并监听按钮的事件）。此时会触发 JS createOrder 函数，并请求到服务器 URL。在该 URL 后台调用 PayPal Create Order API：Create Order API。

调用 Create Order API 时注意，如果需要 3DS 验证，在 Create Order 中传递 payment_source.card.attributes.verification = SCA_ALWAYS；如果不需要 3DS 验证，在 Create Order API 中传递 payment_source.card={}。最后前端返回订单 ID。

Create Order 必传项：

- 商品名称：商品名称必填 purchase_units.items[] https://developer.paypal.com/docs/api/orders/v2/#orders_create!path=purchase_units/items/name&t=request
- 相关跳转 URL 必传：
  payment_source.card.experience_context.return_url
  payment_source.card.experience_context.cancel_url
  https://developer.paypal.com/docs/api/orders/v2/#orders_create!ct=application/json&path=payment_source/card/experience_context&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 5：触发 3DS 验证

如果第四步没有报错，并且在 Create Order API 中传递了 SCA_ALWAYS 参数，3DS 验证窗口会自动触发。

步骤 6：买家 3DS 验证

买家完成 3DS 验证后，会触发前端 JS onApprove 函数。

步骤 7：获取 3DS 返回值

在 onApprove 函数中首先获取 3DS 的返回值 liabilityShift 和订单 ID。

```
onApprove: function (data) {
  const { liabilityShift, orderID } = data;
  if (liabilityShift) {
    /* Handle liability shift. More information in 3D Secure response parameters */
    return fetch(`myserver.com/api/orders/${orderID}/capture`, {
      method: "POST",
    })
      .then((res) => {
        return res.json();
      })
      .then((orderData) => {
        // Redirect to success page
        alert("complete transaction");
      });
  }
},
```

步骤 8：判断 3DS 验证结果

根据 liabilityShift 值判断 3DS 验证是否成功。判断 3DS 参数请参考文档：3DS 验证响应参数

步骤 9：调用 Capture API

如果 3DS 验证成功，请求服务器 URL 调用 PayPal Capture Order API：Capture Order API

步骤 10：处理支付结果

PayPal Capture Order API 是扣款 API，根据 Capture 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 11：处理异步通知：Webhooks

如果 PayPal Capture Order 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

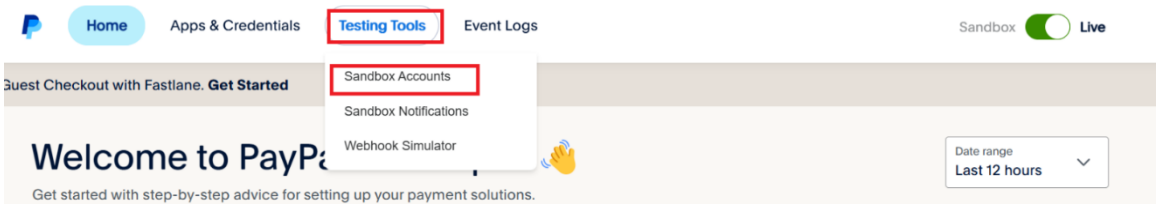
5. 参考文档

- 高级版支付文档: PayPal 高级版支付集成文档
- API 文档: PayPal API 文档
- Server SDK: PayPal Server SDK

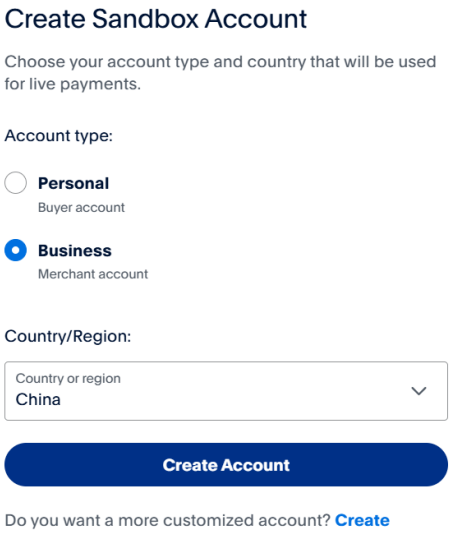
6. 附录

创建商家 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"

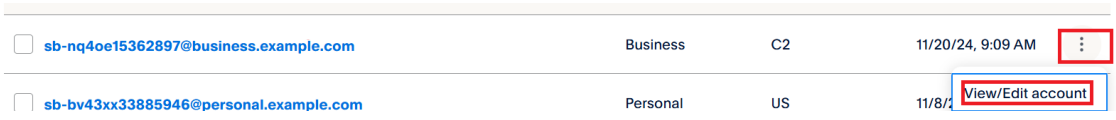


5. 点击 "Create account"
6. 选择 "Business"，选择 "China" 后点击 "Create Account"（如果需要创建 Sandbox 买家账号，在此步骤选择 "Personal"，国家选择非 China。）

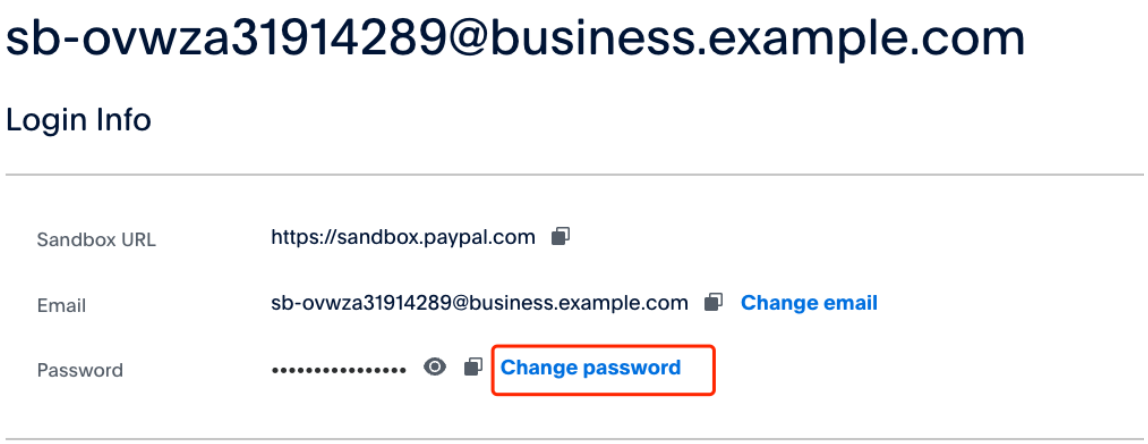


修改 Sandbox 密码步骤

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"

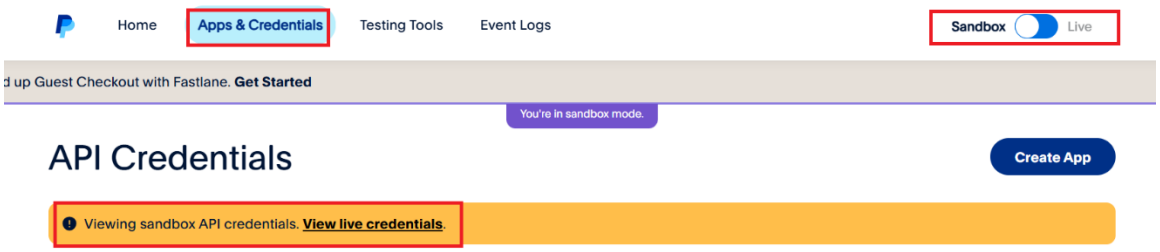


6. 点击 "Change Password" 修改密码



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）

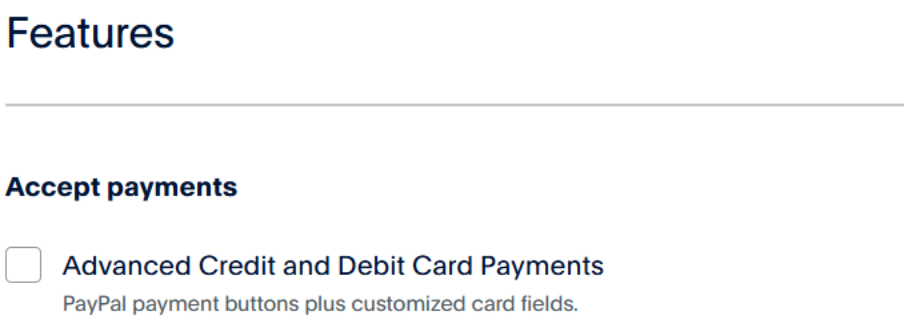


5. 点击 "Create App"
6. 输入 App 名称（只要不重复，可以随意命名），选择上面刚刚创建的 Sandbox Business 账号

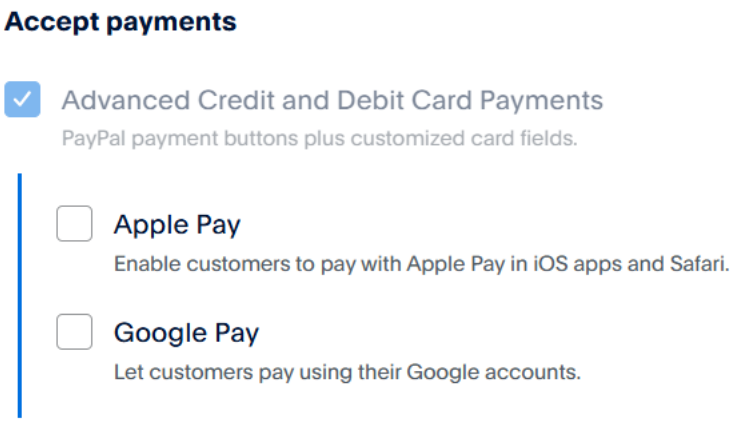
7. 点击 "Create App"

Sandbox 账户开通高级信用卡权限

1. 点击刚刚创建 Rest App Name
2. 勾选 Advanced Credit and Debit Card Payments



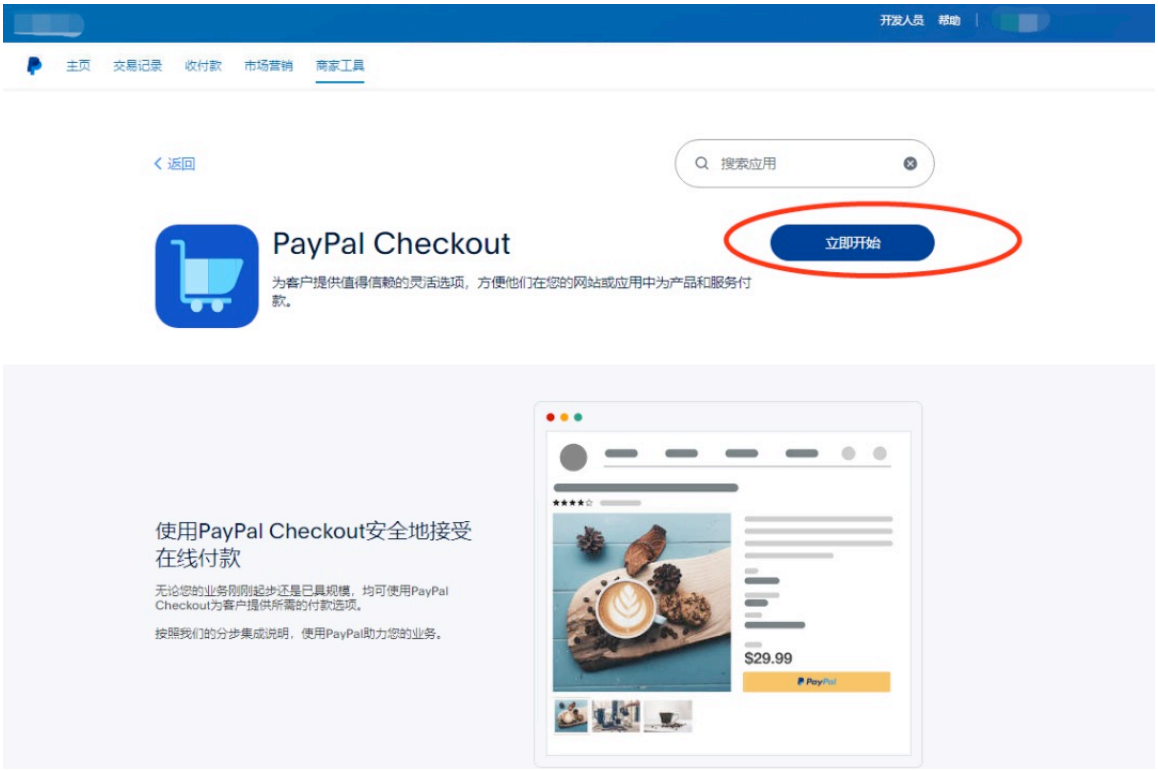
3. 点击 Save Changes
4. 刷新页面查看权限



Live 账户开通高级信用卡权限

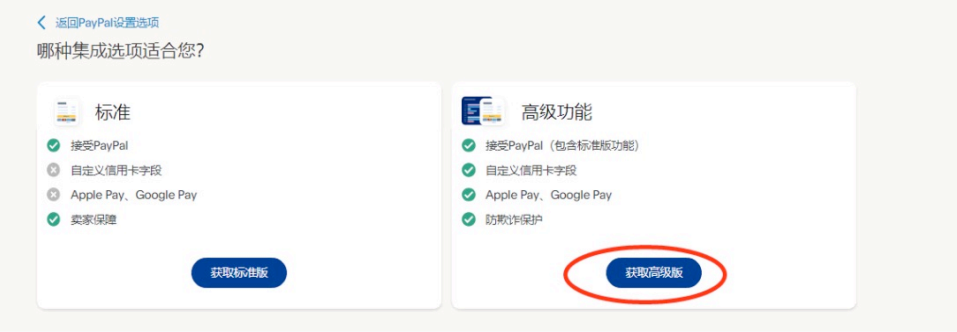
1. 使用真实 PayPal 账号登录 PayPal 主页
2. 点击“商家工具”
3. 点击“PayPal Checkout”

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. 已获准使用高级信用卡