

IERG4120 S24 Tutorial 7

Paypal Integration

Presented by **ZHOU Jiuqin**

Paypal Integration

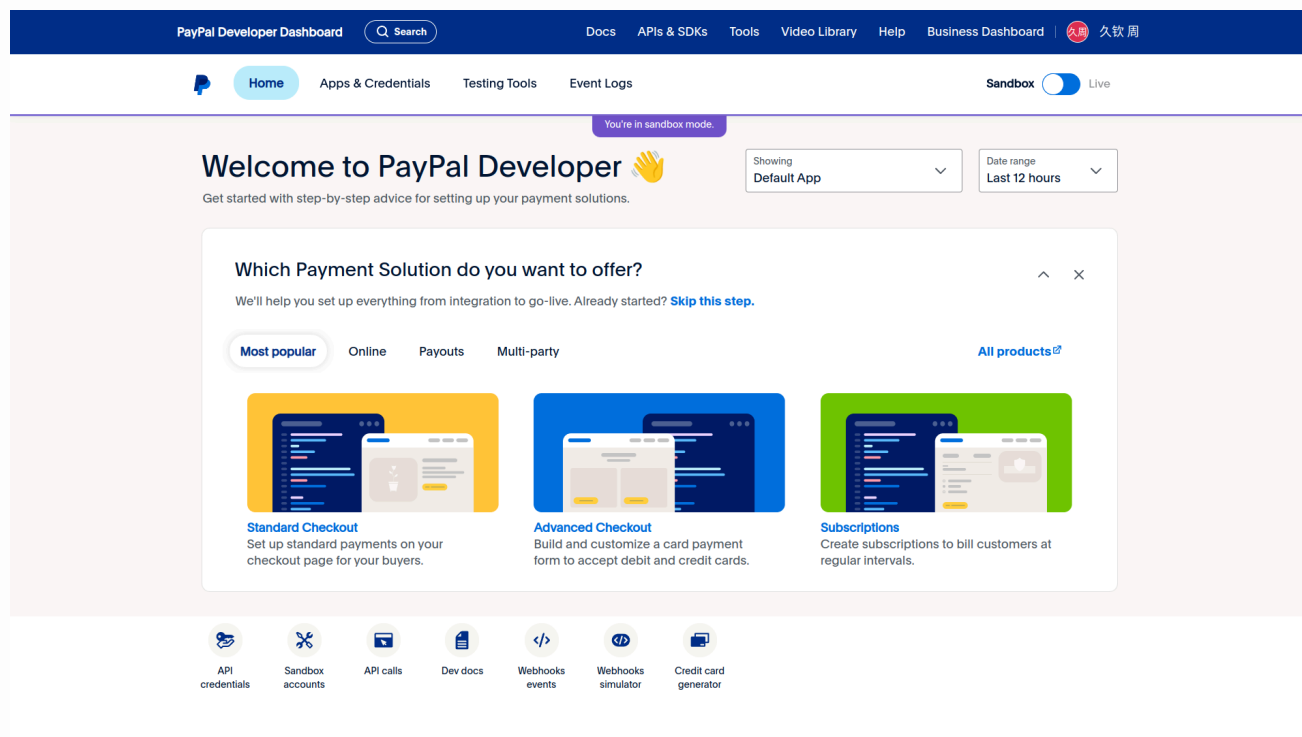
Account Credentials

Checkout Integration

Order Details

Database Management

Common Questions



Sign up and then sign in the paypal developer dashboard at <https://developer.paypal.com/dashboard/>.

Paypal Integration

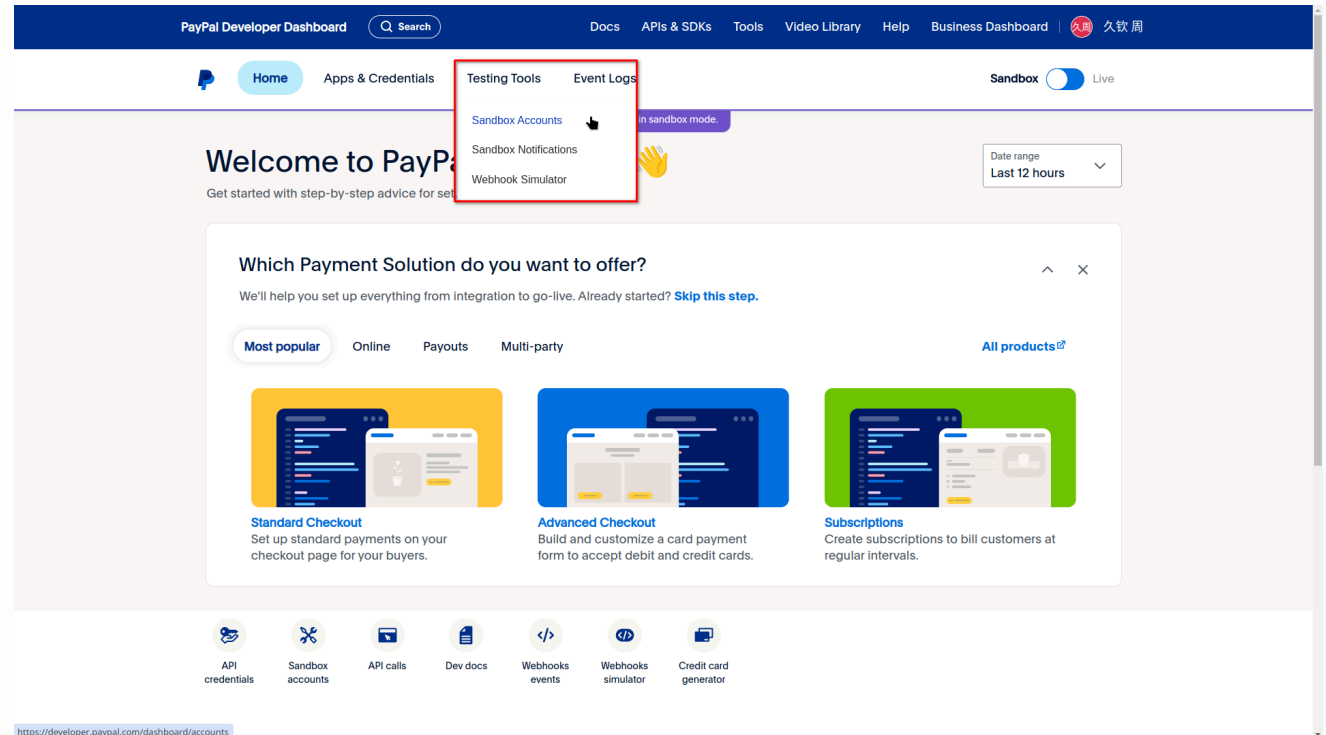
Account Credentials

Checkout Integration

Order Details

Database Management

Common Questions



Click on "Testing Tools" -> "Sandbox Accounts".

Paypal Integration

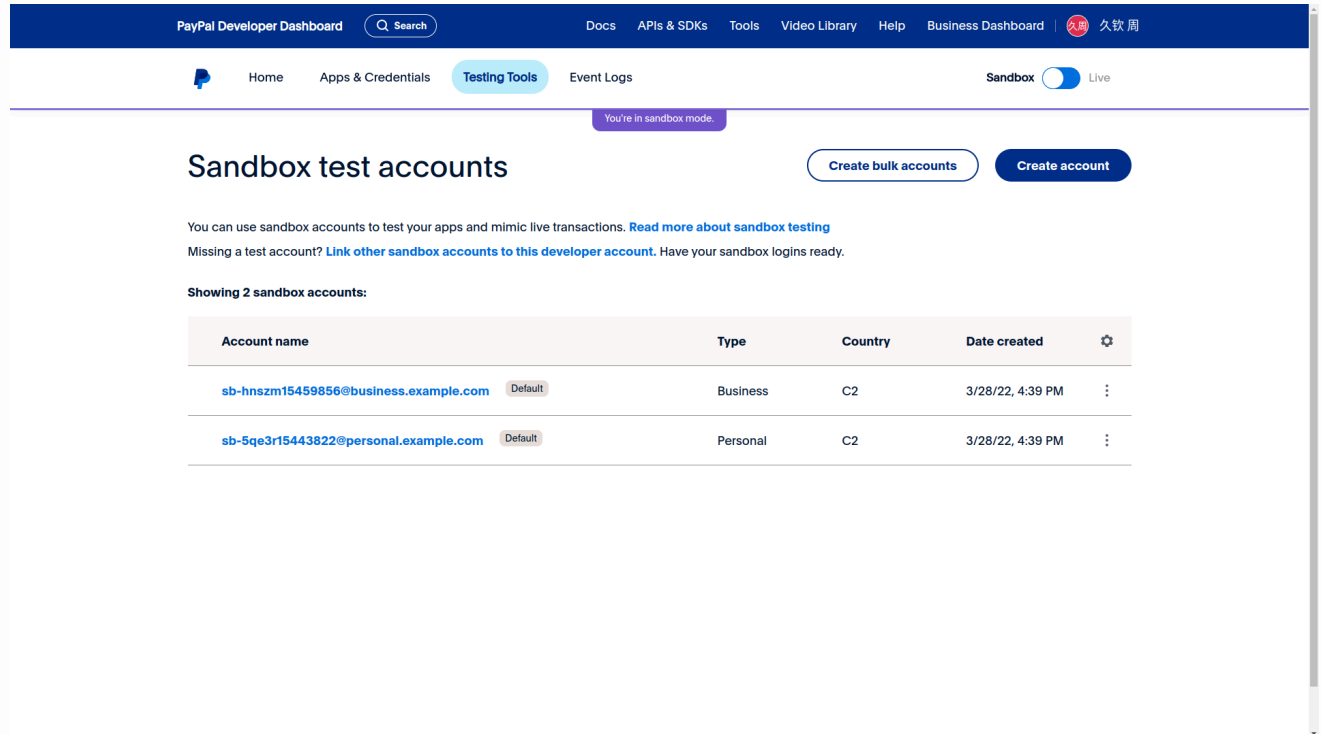
Account Credentials

Checkout Integration

Order Details

Database Management

Common Questions



The screenshot displays the PayPal Developer Dashboard. The top navigation bar includes links for Docs, APIs & SDKs, Tools, Video Library, Help, and Business Dashboard. The main content area is titled "Sandbox test accounts" and features a toggle switch for "Sandbox" (currently on) and "Live". Below the title, there are two buttons: "Create bulk accounts" and "Create account". A message states: "You can use sandbox accounts to test your apps and mimic live transactions. [Read more about sandbox testing](#)". Another message asks: "Missing a test account? [Link other sandbox accounts to this developer account](#). Have your sandbox logins ready." Below this, it says "Showing 2 sandbox accounts:" followed by a table.

Account name	Type	Country	Date created	
sb-hnszm15459856@business.example.com Default	Business	C2	3/28/22, 4:39 PM	⋮
sb-5qe3r15443822@personal.example.com Default	Personal	C2	3/28/22, 4:39 PM	⋮

Two default sandbox accounts have been automatically created.

Paypal Integration

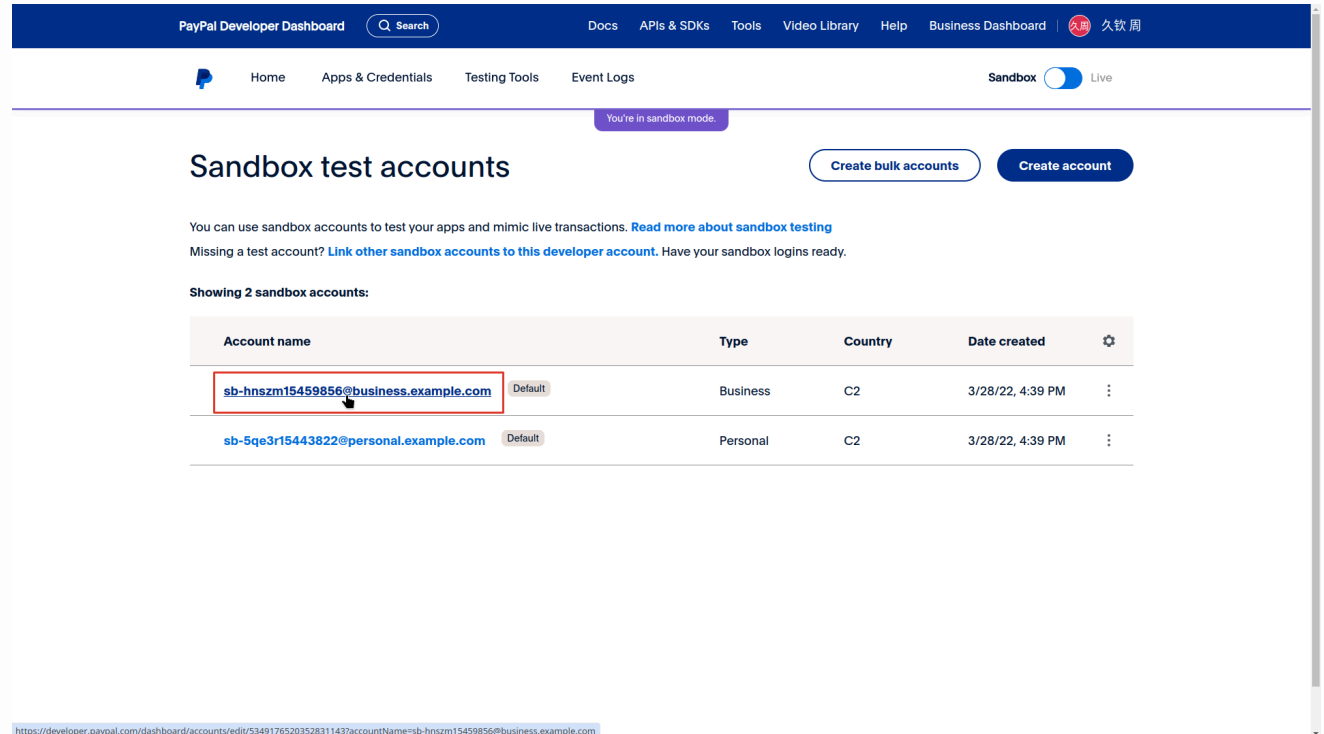
Account Credentials

Checkout Integration

Order Details

Database Management

Common Questions



The screenshot shows the PayPal Developer Dashboard interface. At the top, there's a dark blue header with the 'PayPal Developer Dashboard' title and a search bar. Navigation links include 'Docs', 'APIs & SDKs', 'Tools', 'Video Library', 'Help', 'Business Dashboard', and a user profile '久钦周'. Below this, a secondary navigation bar shows 'Home', 'Apps & Credentials', 'Testing Tools', and 'Event Logs'. A toggle switch for 'Sandbox' is set to 'Live'. A purple banner indicates 'You're in sandbox mode.' The main content area is titled 'Sandbox test accounts' and includes buttons for 'Create bulk accounts' and 'Create account'. It provides instructions on using sandbox accounts and links to 'Read more about sandbox testing'. A table lists two accounts: a business account and a personal account, both created on 3/28/22. The business account is highlighted with a red box and a 'Default' tag. The URL at the bottom is 'https://developer.paypal.com/dashboard/accounts/edit/5349176520352831143?accountName=sb-hnszm15459856@business.example.com'.

PayPal Developer Dashboard

Home Apps & Credentials Testing Tools Event Logs

Sandbox Live

You're in sandbox mode.

Sandbox test accounts

Create bulk accounts Create account

You can use sandbox accounts to test your apps and mimic live transactions. [Read more about sandbox testing](#)

Missing a test account? [Link other sandbox accounts to this developer account](#). Have your sandbox logins ready.

Showing 2 sandbox accounts:

Account name	Type	Country	Date created	
sb-hnszm15459856@business.example.com Default	Business	C2	3/28/22, 4:39 PM	
sb-5qe3r15443822@personal.example.com Default	Personal	C2	3/28/22, 4:39 PM	

<https://developer.paypal.com/dashboard/accounts/edit/5349176520352831143?accountName=sb-hnszm15459856@business.example.com>

Click on the default business accounts.

Paypal Integration

Account Credentials

Checkout Integration

Order Details

Database Management

Common Questions

The screenshot displays the PayPal Developer Dashboard for a sandbox account. The top navigation bar includes links for Docs, APIs & SDKs, Tools, Video Library, Help, and Business Dashboard. The main content area shows the account details for **sb-hnszm15459856@business.example.com**. A red box highlights the login information, which includes the Sandbox URL (<https://sandbox.paypal.com>), the Email (**sb-hnszm15459856@business.example.com**), and the Password (masked with dots). Below this, there is a section for REST API apps, which contains a table with one entry: **Default Application**. The table columns are App name, Client ID, Secret, and Created date. The Secret is also masked with dots. At the bottom, there is a section for NVP/SOAP Sandbox API credentials, showing the Username as **sb-hnszm15459856_api1.business.example.com**.

PayPal Developer Dashboard

Home Apps & Credentials Testing Tools Event Logs

Sandbox Live

You're in sandbox mode.

← Back

sb-hnszm15459856@business.example.com

Login Info

Sandbox URL <https://sandbox.paypal.com>

Email **sb-hnszm15459856@business.example.com**

Password [Change password](#)

REST API apps

App name	Client ID	Secret	Created date
Default Application	Afbrqbxu4lkaCs6pl5-jWj...	3/28/22, 4:39 PM

NVP/SOAP Sandbox API credentials

Username **sb-hnszm15459856_api1.business.example.com**

Email & Password: These are the credentials to login in the business account at the paypal sandbox website <https://sandbox.paypal.com>

Paypal Integration

Account Credentials

Checkout Integration

Order Details

Database Management

Common Questions

The screenshot displays the PayPal Developer Dashboard for a user named 久钦周. The dashboard is in 'Sandbox' mode, as indicated by the toggle switch. The main content area shows the account information for 'sb-hnszm15459856@business.example.com'. Below the login info, there is a section for 'REST API apps' which contains a table with one entry, 'Default Application'. The table columns are 'App name', 'Client ID', 'Secret', and 'Created date'. The 'Client ID' for the 'Default Application' is 'Afbqrqbxu4lkaCs6pl5-jWj...'. Below the table, there is a section for 'NVP/SOAP Sandbox API credentials' with a 'Username' field containing 'sb-hnszm15459856_api1.business.example.com'.

PayPal Developer Dashboard

Docs APIs & SDKs Tools Video Library Help Business Dashboard 久钦周

Home Apps & Credentials Testing Tools Event Logs Sandbox Live

You're in sandbox mode.

← Back

sb-hnszm15459856@business.example.com

Login Info

Sandbox URL https://sandbox.paypal.com

Email sb-hnszm15459856@business.example.com

Password Change password

REST API apps

App name	Client ID	Secret	Created date	
Default Application	Afbqrqbxu4lkaCs6pl5-jWj...	3/28/22, 4:39 PM	

NVP/SOAP Sandbox API credentials

Username sb-hnszm15459856_api1.business.example.com

Client ID: These are the credentials of the associated application in the payment workflows.

Paypal Integration

Account
Credentials

**Checkout
Integration**

Order Details

Database
Management

Common
Questions

Changes to the Front End

Add components to your `<script>` by passing them in the `src` URL using the `client-id` query string parameter.

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

Set up a container element for the button with `id` equals to `paypal-button-container`.

```
<div id="paypal-button-container"></div>
```


Paypal Integration

Account
Credentials

**Checkout
Integration**

Order Details

Database
Management

Common
Questions

Then, customize your buttons using the `style` option.

```
1 paypal.Buttons({  
2   style: {  
3     layout: 'vertical',  
4     color: 'blue',  
5     shape: 'rect',  
6     label: 'paypal'  
7   }  
8 }).render('#paypal-button-container');
```

Alternatively, ignore lines 2-7 and the default styles will be applied.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions


The button on your website will look like the following:



Web

 **PayPal**

 **Pay Later**

 **Debit or Credit Card**

Powered by **PayPal**

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

Action `createOrder`: The function sets up the details of the transaction. Pass it as a parameter in `paypal.Buttons`.

```
1 paypal.Buttons({
2   createOrder: async (data, actions) => {
3     let orderDetails =
4       await fetch("/my-server/get-order-details", {
5         method: "POST",
6         headers: {
7           "Content-Type": "application/json",
8         },
9         body: JSON.stringify({ /* Cart Details */ })
10      })
11      .then((response) => response.json());
12      return actions.order.create(orderDetails);
13    }
14  }).render('#paypal-button-container');
```

Paypal Integration

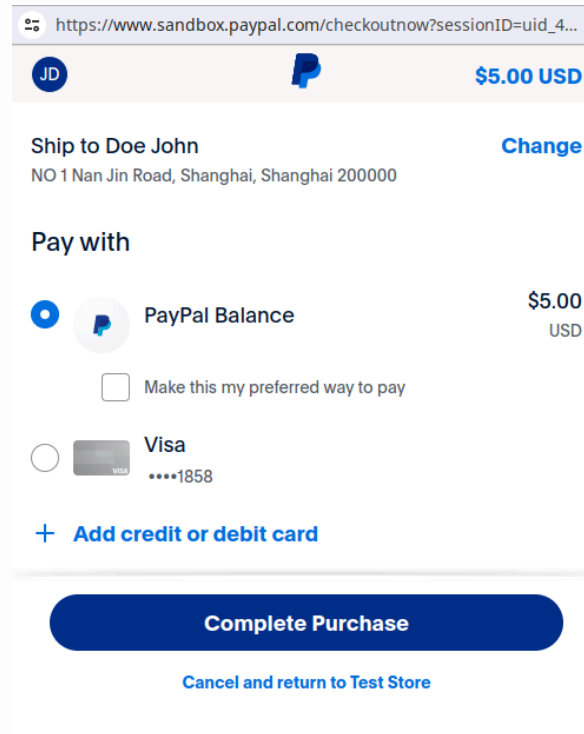
Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions



The screenshot shows a PayPal checkout page from a sandbox environment. The URL in the browser is `https://www.sandbox.paypal.com/checkoutnow?sessionID=uid_4...`. The page header includes a user icon 'JD', the PayPal logo, and the amount '\$5.00 USD'. The shipping address is 'Ship to Doe John, NO 1 Nan Jin Road, Shanghai, Shanghai 200000', with a 'Change' link. Under the 'Pay with' section, 'PayPal Balance' is selected with a radio button, showing '\$5.00 USD'. There is an unchecked checkbox for 'Make this my preferred way to pay'. Below it, a Visa card is listed with a masked number '****1858'. A link '+ Add credit or debit card' is present. At the bottom, there is a large blue 'Complete Purchase' button and a smaller link 'Cancel and return to Test Store'.

When the buyer selects the PayPal button, `createOrder` launches the PayPal Checkout window. The buyer logs in and approves the transaction on the `paypal.com` website.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

Action onApprove: It captures the funds from the transaction and shows a message that tells the buyer the payment was successful.

```
1  onApprove: async (data, actions) => {  
2    return actions.order.capture()  
3      .then(async (orderDetails) => {  
4        await fetch("/my-server/save-order", {  
5          method: "POST",  
6          headers: {  
7            "Content-Type": "application/json",  
8          },  
9          body: JSON.stringify(orderDetails)  
10         });  
11       });  
12   };
```

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

Action onCancel: It is called when a buyer cancels a payment, showing a cancellation page or returning to the shopping cart.

```
paypal.Buttons({  
  /* createOrder omitted. */  
  /* onApprove omitted. */  
  onCancel: (data) => {  
    // Show a cancel page, or return to cart  
  }  
}).render('#paypal-button-container');
```

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

Action onError: If an error prevents buyer checkout, alert the user that an error has occurred with the buttons using the callback.

```
paypal.Buttons({  
  /* createOrder omitted. */  
  /* onApprove omitted. */  
  /* onCancel omitted. */  
  onError: (err,) => {  
    // Redirect to a specific error page  
  }  
}).render('#paypal-button-container');
```

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

Changes to the Back End

```
1 paypal.Buttons({  
2   createOrder: async (data, actions) => {  
3     let orderDetails =  
4       await fetch("/my-server/get-order-details", {  
5         method: "POST",
```

A Recap on the action createOrder

URL	API Description
/my-server/get-order-details	This API accepts <code>cartDetails</code> and returns <code>orderDetails</code> in a format accepted by PayPal.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

```
1  /* createOrder omitted */
2  onApprove: async (data, actions) => {
3    return actions.order.capture()
4      .then(async (orderDetails) => {
5        await fetch("/my-server/save-order", {
6          method: "POST",
7          headers: {
```

A Recap on the action onApprove

URL	API Description
/my-server/save-order	This API accepts orderDetails if digest matches and then stores the order in the ORDERS database on the server.

The concept of **digest** will be covered later in Database Management.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

```
1 paypal.Buttons({
2   createOrder: async (data, actions) => {
3     let orderDetails =
4       await fetch("/my-server/get-order-details", {
5         method: "POST",
6         headers: {
7           "Content-Type": "application/json",
8         },
9         body: JSON.stringify({ /* Cart Details */ })
10      })
11      .then((response) => response.json());
12      return actions.order.create(orderDetails);
13    }
14  }).render('#paypal-button-container');
```

A Recap on orderDetails in the action createOrder

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

```
1 {  
2   "purchase_units": [  
3     {  
4       "amount": /* Omitted Here */,  
5       "items": [ /* Omitted Here */ ],  
6       "custom_id": /* Omitted Here */,  
7       "invoice_id": /* Omitted Here */  
8     }  
9   ]  
10 }
```

An example of `orderDetails` (JSON) required to create orders.

Property	Description
<code>purchase_units</code> required	Array of objects non-empty An array of purchase units. Each purchase unit establishes a contract between a customer and merchant.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

REQUEST BODY SCHEMA

```
1 {  
2   "purchase_units": [ /* Omitted Here */ ]  
3 }
```

Property	Description
purchase_units required	Array of objects non-empty An array of purchase units. Each purchase unit establishes a contract between a customer and merchant.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

```
1      "amount": /* Omitted Here */,
2      "items": [ /* Omitted Here */ ],
3      "custom_id": /* Omitted Here */,
4      "invoice_id": /* Omitted Here */
```

Property	Description
amount required	object The amount to collect.
items	Array of objects An array of items that the customer is purchasing from the merchant.
custom_id	string <= 127 characters The client-provided external ID. Used to reconcile client transactions with PayPal transactions. Returned in transaction and settlement reports. Only supported for the PayPal payment method.
invoice_id	string <= 256 characters The API caller-provided external invoice ID for this order.. Only supported for the PayPal payment method.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

```
1      "amount": {
2          "currency_code": "USD",
3          "value": 7,
4          "breakdown": {
5              "item_total": {
6                  "currency_code": "USD",
7                  "value": 7
8              }
9          }
10     }
```

Property	Description
currency_code required	string The three-character ISO-4217 currency code . PayPal does not support all currencies.
value required	string The total amount charged to the payee by the payer.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

```
"items": [  
  {  
    "name": "ProductA",  
    "unit_amount": {  
      "currency_code": "USD",  
      "value": 1  
    },  
    "quantity": 1  
  }  
]
```

Property	Description
name	string
required	The item name. Maximum length is 127 characters.
quantity	string <= 10 characters
required	The item quantity. Must be a whole number.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

**Database
Management**

Common
Questions

Recommended Database Schema

Name	Type	Remarks
UUID	TEXT	PRIMARY KEY
username	TEXT	
digest	TEXT	
salt	TEXT	
orderDetails	TEXT	

Columns in Database Affected by API

URL	Columns Affected
/my-server/create-order	UUID, username, digest, salt
/my-server/save-order	orderDetails
/my-server/cancel-order	ALL

Account
Credentials

Checkout
Integration

Order Details

**Database
Management**

Common
Questions

Meaning of Digest

/my-server/create-order calls genDigest to generate a hash (**Digest**) of an string composed of at least:

1. the pid and quantity of each selected product
2. The current price of each selected product
3. The total price of all selected products
4. Currency code
5. Merchant's email address
6. A random salt

The digest provides a way to ensure that **all variables above are not modified by clients or third parties**. Since the reverse of a hash is difficult, if two hash (the one you sent and the one you received) are the same, you can assume that the above variables are the same.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

**Common
Questions**

Question 1

Explain more about the creation and settings of sandbox accounts if default accounts are not created automatically.

Answer: First Sign up for a real Paypal account, log into [Sandbox accounts - PayPal Developer](#), and create two test accounts by clicking the blue bottom “Create Account”. One account is a merchant account, and the other is a buyer account.

Then go to [Applications - PayPal Developer](#), a default application should have been created for the business type account in the sandbox. Click on it to view details, save the Client ID, and it will be useful when using the client-side javascript code for checkout.

Paypal Integration

Account
Credentials

Checkout
Integration

Order Details

Database
Management

Common
Questions

Question 2

After the payment, the Paypal says that “The transaction has already been paid.”, or it returns “Error: Api: /smart/api/order/xxxx/capture returned status code: 500 (Corr ID: xxxx)” in the browser console?

Answer: This happens when you pass the duplicate invoice_id to the same merchant_account of Paypal. You need to ensure that invoice_id is unique for every transaction. See [Universally unique identifier - Wikipedia](#) for the design of non-conflict UUID.

You can turn off payment blocking temporarily at [Preferences - PayPal](#) of your merchant account for debugging use. In the final version of your code, lastInsertId() (or a UUID generated based on it) should work properly to return a unique value for each transaction.