**PayPal Integration in PWA Kit**

* So, let’s start PayPal integration in PWA kit (0-5 sec)
* First of all, you need to create **PayPal Developer Account.** (6-9sec)
* For that go to [developer.paypal.com](https://developer.paypal.com/home) (10-14sec)
* Here **Log in to Dashboard** ,if you do not have account then create it and then login to the dashboard (14-23sec)
* After successful login, go to **My Apps & Credentials** (26-30sec)
* Here first of all, you need to create an app, click on **Create App** (31-36sec)
* Here give the App Name, App type and Sandbox Business Account after that click on the **Create App** (40-49sec)
* After creating the app, you will see your app in the list, click on it (51-57sec)
* [After Opening App Detail Page] here you will get the **client id** and **client secret,** we will use these in our code (1:01-1:08)
* So, after creating the app we need to do changes in our code (1:11-1:16)
* So, first of all, we need to change **ssr.js** file (1:17-1:21)
* [going to **ssr.js file**] In **ssr.js file**, in **connect-src** we need to give the domain names of the api (1:24-1:34)
* [pointing to the first domain] So, this domain is for calling the first api that will redirect the user to the **PayPal Gateway** [pointing to the second domain] and this second is for the second api which will capture the payment. (1:35-1:48)
* After this, we need to do the changes in the **payment-selection.jsx** file (1:51-2:01)
* [going to the **payment-selection.jsx file**] So, basically this file renders the payment section in the checkout page (2:02-2:10)
* [going to the checkout page in react app] So, when we go to the react app here, So basically this is the checkout page [pointing to the payment section of checkout page] and this particular payment section is rendered by that **payment-selection**
* file (2:11-2:27)
* [going to the payment-selection.jsx file] [pointing to the radio group component]so, in this we are using the radio group component that we are importing from the **ChakraUI** (2:30-2:39)
* [going to the documentation] You can get the link of **that** in here, just click on it and you will be redirected to the ChakraUI **Radio Component** (2:41-2:53)
* [after redirecting to the ChakraUI radio component page] So, you can check here the usage of these components. Also, you can use different components that are in here just by importing it in the code.(2:54-3:05)
* [going to the payment-selection.jsx file] So in the code here, we are using **radio group component** and we are calling the **handleSelection** function on the **onChange** event (3:09-3:20)
* So, whenever we switch the radio buttons, then this **handleSelection** function will be called.(3:20-3:31)
* [going to the **handleSelection** function] So in **handleSelection,** we are checking if the value of the radio button is PayPal so we are setting the value to the PayPal so that it will switch the radio button to the PayPal payment method (3:32-3:51)
* [pointing to the **orderTotal** variable] So in the variable **orderTotal,** we are storing the total of the order from the basket object (3:52-4:00)
* And if we are selecting **PayPal,** then we are calling the first api (4:02-4:08)
* So, these are the headers, first of all we are encoding the **client id** and **secret** to **base64** here like this [pointing to the **base64encodedkey** variable] and we are setting these headers as provided in the documentation (4:09-4:27)
* [pointing to the body of api] and this is the body of the api that we are going to configured in the api (4:28-4:35)
* [pointing to the **orderTotal** in the body of the api] and here we are sending the total of the order so that it will be shown in the payment gateway after redirecting.

(4:37-4:49)

* [pointing to the **return\_url** and **cancel\_url** in **body**] and this is the **return\_url** and **cancel\_url,** so if the payment is approved then it will redirect to our **return\_url** hereand if the payment is not approved then it will redirect to the **cancel\_url** which is here[pointing to the **cancel\_url**] (4:52-5:11)
* So after calling the api, if the api is successful then we are storing the **capture url** which we are getting from the response of the api in the **captureURL** variable.

(5:12-5:31)

* And then we are storing the **captureURL** in our local storage so that we can use it in our second api call to capture the payment (5:32-5:43)
* Also, we are getting the **redirect url** so that we can redirect to the payment gateway, so using this **redirect url** we are redirecting to the payment gateway using this (5:44-5:59)
* And then if the payment method is not PayPal then we are setting it to **Credit Card** (6:00-6:11)
* [going to the checkout page of retail app] So when we go to our retail app, so if we are selecting this **PayPal** then it will redirect us to the payment gateway (6:12-6:23)
* [After redirecting to the payment gateway] So here we need to login with the **Payer’s Account** so(6:24-6:31)
* [going to the PayPal developer dashboard and then copy the Email and Password of Payer’s Account]
* [going back to payment gateway and Paste Email and Password in the respected field and then Login]
* [After Login] [pointing to the total amount] so here will be the order total that we are passing in the body and then **Continue,** [click on continue] so it will redirect us to the **return url** that we are passing to the body of the first api call (7:07-7:24)
* [After redirecting to the **return url**] so we will get the **PayerID** herein the parameter and also the **status** which is approved, so we will use these in our second api call (7:38-7:53)
* Next we have to do changes in our **checkout-context.js** (8:00-8:06)
* [going to the **checkout-context.js file**] So, in this we are checking the URL and in URL we are getting the **status** (8:07-8:14)
* [going to the checkout page on which we are redirected to by the payment gateway] which is here, this [pointing to the status parameter in URL] (8:14-8:17)
* [going back to **checkout-context.js**] So we are checking if the status is not approved then we are setting the payment instrument to **Credit Card** (8:20-8:29)
* And if the payment is approved then we are setting the payment instrument to **PayPal,** so it will set instrument to PayPal (8:30-8:40)
* [going to the index.jsx in app/pages/checkout/index.jsx] And lastly in **index.jsx** which is in the checkout component of the pages. (8:42-8:48)
* [going to **index.jsx file** in app/pages/checkout/index.jsx] here,this file renders the **Place Order** button of the checkout page (8:50-8:55)
* [going to the final checkout page] just as here, so this button is rendered by the checkout **index.jsx** [pointing to the **Place Order** button] (8:56-9:04)
* [going back to index.jsx file] this file, so on clicking on this [pointing to the **Place Order** Button code], it will call this **submitOrder** function (9:05-9:14)
* [going to the **submitOrder** function] And in **submitOrder** we are calling a **customCheckout** function which we have created for PayPal Integration (9:15-9:23)
* [going to the **customCheckout** function] In **customCheckout,** we are checking the status and we are fetching the **PayerID** that we are getting from the **redirect url** (9:27-9:38)
* [going to the final checkout page] here,So we will pass this **PayerID** in the body of the second api call [pointing to the **PayerID** parameter of the URL] (9:39-9:44)
* [going back to index.jsx file] And we are checking if the **paymentStatus** is approved, if it is approved then we have to call the second api to capture the payment

(9:49-9:58)

* [pointing to the **captureUrl** variable] So here we are taking the capture url that we have stored in the first api call (10:00-10:05)
* [pointing to the **base64encodedkey** variable] and then this is the encoded **client id** and **secret** in **base64** (10:08-10:14)
* [pointing to the headers] And we are setting the headers again as referred in the documentation (10:15-10:19)
* [pointing to the **PayerID** in the body of the api] This is the **PayerID** we are passing in the body and then we are calling the api (10:22-10:29)
* So, if the api is successfully called then the payment is captured successfully and we are returning **true** (10:30-10:39)
* Else if there is any error then it is returning **false** (10:40-10:43)
* [pointing to the **isCaptured** variable in **submitOrder** function] So we are checking the **isCaptured** variable, so if the payment is captured then it will be **true** and if it is not captured then it will be **false** (10:45-10:56)
* So, if the payment is captured then we will place the order and then we will navigate to the confirmation page (10:57-11:04)
* [going to the checkout page of retail app] So when we go to our retail app, when we click on **Place Order** [Click on place order button] [After redirecting to the confirmation page] so it will redirect to our confirmation page and also the payment is captured also. (11:05-11:22)
* [going to [www.sandbox.paypal.com](https://www.sandbox.paypal.com/signin) ]So, when we go to our Payer’s account, So we have to login in the [www.sandbox.paypal.com](https://www.sandbox.paypal.com/signin) with the **Payer’s Account credentials** so (11:23-11:37)
* [Going to to [PayPal Developer](https://developer.paypal.com/home) page and then **login to dashboard**]
* [After logging to dashboard, go to **Accounts** then click on the three **dots** of whose **Payer’s Account Credentials** you want and click on **View/Edit account** and then copy the **Email** and **Password** of the **Payer’s account**]
* [Going back to PayPal Sandbox Login and paste the **Payer’s Account credentials** in respected fields and then click on **Login**]
* [After Successful Login] So, when we login we will see that the payment is deducted from the payer’s account [click on **Log Out**] (12:04-12:10)
* [Click on Login] And when we login to our business account (12:14-12:18)
* [go to [PayPal Developer](https://developer.paypal.com/home) page and then **login to dashboard]**
* [After logging to dashboard, go to **Accounts** then click on the three **dots** of **Business Account Credentials** and click on **View/Edit account** and then copy the **Email** and **Password** of the **Business account**]
* [Going back to PayPal Sandbox Login and paste those **Business account credentials** in respected fields and then **Login**]
* [After Successful Login] Then we will see that the transaction is carried and the payment is credited in our **account,** So the payment is captured (12:46-12:57)
* [going to our Commerce Cloud Sandbox and after logging in] Also in our sandbox, we have to check whether the order is placed or not (12:58-13:04)
* [pointing to the recently placed order] So, we will see that the order is also placed with the id given in the retail app. [going to retail app] (13:42-13:48)

**THANK YOU** (13:51-13:52)