WPEC Helpful Code Pointers/Comments

# Default Settings at Plugin Activation Time

#) The following function is run at plugin activation

Main::single\_activate()

The following function can be used to set the default settings values:

Main::get\_defaults()

# PayPal API Code Related

=================================

#) The following function shows how the HTML code is output for the button using a shortocde:

WPEC Main -> public/includes/class-shortcodes.php/generate\_pp\_express\_checkout\_button()

#) WPEC Main -> the "wpec\_process\_payment" function gets triggered from the payment JS "onApprove" action. It then takes the data from the request and adds to the database to populate the orders menu.

1) WPEC Main -> includes/class-process-subscription.php -> get\_plan\_id() (the "wpec\_sub\_get\_plan\_id" ajax hook)

2) [webhook create] Follow the breadcrumbs of 'wpec\_sub\_create\_webhook' ajax request (from the "Create Webhook" button) so you can see how a webhook is created for the site.

3) [webhook notification handling] The following function handles the webhooks that arrive on the URL. It simply listens for REQUEST with a particular query argument that we set earlier.

WPEC-sub-addon -> webhook-handler/class-listener.php -> handle\_webhook()

Adding some extra logs in that function will be helpful

4) Follow the breadcrumbs of add\_event() function and 'wpec\_sub\_events' post\_meta to see how all the subscription related webhook events are saved in the post meta.

5) Use the following function to see how subscription is added to the DB.

WPEC-sub-addon -> class-process-subscription.php/add\_subscription()

6) The following breadcrumbs show how a NEW subscripiton checkout is added to the DB.

When the transaction completes on the popup: JS code -> "handler.buttonArgs.onApprove"

Then it goes to the folloiwng function on the server side: class-process-subscription.php -> add\_subscription( $order, $payment, $data )

6.1) Search for the following key actions in the JS files (or the plugin) to see how/what parameters are used to create an order or a subscripiton payent/order from the front-end checkout JS code.

buttonArgs.createOrder (used for one time buy now buttons)

buttonArgs.createSubscription (used for recurring/subscription payment buttons)

buttonArgs.onApprove

## PayPal API Call - Create-Order

--------------------------------------------------

The data from the create order call is set/massaged in the following function. Search for it to find the responsible code:

PayPal\_Payment\_Button\_Ajax\_Handler::create\_order\_data\_for\_pp\_api()

You can also search for the following code which we use for create-order:

        //Create the request for the PayPal API call.

        $request = new Request( '/v2/checkout/orders', 'POST' );

        $request->body = $order\_data;

        //Execute the request.

        $client = Client::client();

        $response = $client->execute($request);

//Alternative way to create the order using the PayPal SDK (the above code is the same as this one below)

//Need to add this to the file - use PayPalCheckoutSdk\Orders\OrdersCreateRequest;

// $request = new OrdersCreateRequest();

// $request->body = $order\_data;