An Introduction Page
for
PaypalIPN.php Script

October 2nd 2022: I am trying to establish notes per references in the code.

DOC - CONSTANT SECTION

Changed the way CONSTANTS are created. Initially with const, but now using DEFINE allows them to be dynamic, which allows this script to manage LOCAL vs. SIMULATOR testing from one variable in the paypal-ipn.php script. Oct 2nd 2022 this seems to be working.

DOC - CHANGE LOG

Generally changes in the code are noted here.

DOC - BEGIN ALL CLASS CODE

Noticed another use of const, so will have to change this code in the near future.

DOC - ECHO

echo is used through the code to display information in the LOCAL TEST MODE to the screen. These can stay in place for DEV, but will likely be removed when packaged.

Document for PaypalIPN.php

```
<?php class PaypalIPN
/**
* 09-12-12 MAJOR CODE CHANGES FOR CONSTANT DEFINING
*   Need constants  1)  public function set_name:
*                       Defined in the running code,
*                       so must use "define" for constants.
*                   2)  public function get_name:
*                       Get a 0 or 1 via if statements
*                       if (..) ){define two CONSTANTS}
*                       else {define two CONSTANTS the opposite
way}
*                   3)  Now in main block of code call the
constant a new way:
*                       if ($res == self::VALID) {
*                       replaced with:
*                       if ($res == VALID) {
*                           by definition of "define" the
CONSTANT
*                           it is visible EVERYWHERE.
*                   4)  Therefore, 1) Can define from parent
script = paypal_ipn.php
*                       variable: $test_local = 0;   // 1=
Local, 0 = Paypal Simulator
*                   5)  Thus put these functions inside this
CLASS and
*                       just Below: const SANDBOX_VERIFY_URI =
*                   6)  Now LOCAL and PP IPN SIMULATOR can be
managed with one
*                       variable on line approx line 65 of
paypal-ipn.php script!
*                   THE END OF DOC ON THIS TOPIC
*
* 09-27-22 - echo print lines have been left in for local
debugging & do not affect PP Simulator.
* 09-27-22 - DVA re-testing locally, tested Paypal Simulator.
Both Good!
* 09-26-22 - DVA re-testing locally
* 09-09-22 @8:31pm - DVA This is SIMULATOR USE - SANDBOX
* 09-09-22 5pm DVA -Starting IPN Simulator Testing
* Simulator URL Used = https://www.wpappsforthat.com/index.php/
paypal-ipn/
*   PP Said: IPN was sent and the handshake was verified.
*   DVA commenting out all ECHO STATEMENTS
* 09-09-22 DVA - Now writing to a text file. Still must add DB +
test various values for type of transaction
```

```
* 09-02-22 DVA - Testing down to line 107 and then seems to call
ipn (other script)
* 08-28-22 DVA - this code from here on github, it seems like
latest
* https://github.com/paypal/ipn-code-samples/blob/master/php/
PaypalIPN.php
* Maybe) Latest commit cb738c4 on Mar 16, 2018
* 09-03-22 -> https://stackoverflow.com/questions/14015144/
sample-php-code-to-integrate-paypal-ipn
*/
    // ******************** //
{   // BEGIN ALL CLASS CODE //
    // ******************** //

    /** @var bool Indicates if the sandbox endpoint is used. */
    //private $use_sandbox = false;
    private $use_sandbox = true;
    /** @var bool Indicates if the local certificates are used.
*/
    private $use_local_certs = true;

    /** Production Postback URL */
    // 09-7-22 think only one can be used
    //const VERIFY_URI = 'https://ipnpb.paypal.com/cgi-bin/
webscr';
    /** Sandbox Postback URL */
    const SANDBOX_VERIFY_URI = 'https://
ipnpb.sandbox.paypal.com/cgi-bin/webscr';

    /** Response from PayPal indicating validation was
successful */
        // 09-07-22 Per PP https://developer.paypal.com/api/nvp-
soap/ipn/IPNTesting/#local-development-testing
        //  local testing flip-flop code logic, but I am
        //  un-sure if do it here. So try here and see if
        //  we get IPN-8b echoed. If so, may be going the right
direction.

    // 09-07-22 see URL and discussion below: https://
developer.paypal.com/api/nvp-soap/ipn/IPNTesting/ #local-
development-testing

    // GET VALUE FOR SIMULATOR USE - LOCAL or IPN SIMULATOR //

        // Properties
        public $test_location;
    // Methods - got passed setting from var = $test_local =
0 | 1
    public function set_name($test_location) {
      $this->test_location = $test_location;
    }

    // Get value from above function to determine LOCAL or
IPN SIMULATOR test condition

    function get_name ($test_location) {
echo "<br>IPN-1a1-> Inside get_name function: $this-
>test_location<br>";
      // LOCAL - DEFINE CONSTANTS //
      if ($this->test_location ==1) {
        define("VALID", "INVALID");
        define("INVALID", "VERIFIED");
        $sim_verified = "";
        $sim_invalid = "";
        $local_verified = "INVALID";
        $local_invalid = "VERIFIED";
echo "IPN-1a2-> Invalid = ";
echo INVALID;
echo "<br>IPN-1a3-> <b> <FONT COLOR='green'>RUN LOCAL TEST</
font></b><br>";
      }
      else {
        // PP SIMULATOR - DEFINE CONSTANTS //
echo "<br>IPN-1b1->USE SIMULATOR IPN TEST<br>";
        define("VALID", "VERIFIED");
        define("INVALID", "INVALID");
        $local_verified = "";
        $local_invalid = "";
        $sim_verified = "VERIFIED";
        $sim_invalid = "INVALID";
echo "IPN-1b2-> Valid = ";
echo VALID;
echo "<br>IPN-1b3-> <FONT COLOR='red'>USE SIMULATOR TEST</
font><br>";
              }
      return array ($local_verified, $local_invalid,
$sim_verified, $sim_invalid);
    }
  // ****************************************************** //

  /**
```

```
     * Sets the IPN verification to sandbox mode (for use when testing,
     * should not be enabled in production).
     * @return void
     */
    public function useSandbox() {
        $this->use_sandbox = true;
    }

    /**
     * Sets curl to use php curl's built in certs (may be required in some
     * environments).
     * @return void
     */
    public function usePHPCerts() {
        // 09-04-22 Using certs from wpappsforthat.com
        $this->use_local_certs = false;
        //$this->use_local_certs = true;
        // if req'd using = "/home1/davelkwd/wpappsforthat.com/wp-includes/certificates/ca-bundle.crt"
    }

    /**
     * Determine endpoint to post the verification data to.
     *
     * @return string
     */
    public function getPaypalUri() {
        if ($this->use_sandbox) {
            return self::SANDBOX_VERIFY_URI;
        } else {
            return self::VERIFY_URI;
        }
    }

    /**
     * Verification Function
     * Sends the incoming post data back to PayPal using the cURL library.
     *
     * @return bool
     * @throws Exception
     */

    public function verifyIPN() {
        //$this->test_location = $test_location;
        //die;
        if ( ! count($_POST)) {
            throw new Exception("Missing POST Data");
        }
echo "<br>IPN-1c1-> Valid = ";
echo VALID;
echo  "<br>";
echo "IPN-1c2-> Invalid = ";
echo INVALID;
echo  "<br>";

        $raw_post_data = file_get_contents('php://input');  //
09-02-22 Seems to want datae. Time to pass real data per example
from PP
        $raw_post_array = explode('&', $raw_post_data);
        $myPost = array();
        foreach ($raw_post_array as $keyval) {
            $keyval = explode('=', $keyval);
            if (count($keyval) == 2) {
                // Since we do not want the plus in the datetime
string to be encoded to a space, we manually encode it.
                if ($keyval[0] === 'payment_date') {
                    if (substr_count($keyval[1], '+') === 1) {
                        $keyval[1] = str_replace('+', '%2B',
$keyval[1]);
                    }
                }
                $myPost[$keyval[0]] = urldecode($keyval[1]);
            }
        }

        // Build the body of the verification post request,
adding the _notify-validate command.
        $req = 'cmd=_notify-validate';
        $logfile = "";

        // really need to save to DB here
        foreach ($myPost as $key => $value) {
            $req .= "&$key=$value";
        }

        // ****************************  //

        // 09-09-22 DVA REVIEW THIS PP CODE FOR CURL lines
https://developer.paypal.com/api/nvp-soap/ipn/ht-ipn/
```

```
        //          May have to page down a way.
        //          Also review all that code (even though old)
it may have some interesting info to make better code.
        // Post the data back to PayPal, using curl. Throw
exceptions if errors occur.
        $ch = curl_init($this->getPaypalUri());
        curl_setopt($ch, CURLOPT_HTTP_VERSION,
CURL_HTTP_VERSION_1_1);
        curl_setopt($ch, CURLOPT_POST, 1);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch, CURLOPT_POSTFIELDS, $req);
        //curl_setopt($ch, CURLOPT_SSLVERSION, 6);    // Not
used by AngellEYE PaypayIPN
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 1);
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);

        // DVA: NOT EVEN SURE IF USED / REQUIRED: This is often
required if the server is missing a global cert bundle, or is
using an outdated one.
        //if ($this->use_local_certs = "true") { // 09-27-22 was
returning invalid result, so change to match above if correct
        if ($this->use_local_certs = "true") {
echo "IPN-1d1-> Use local cert = $this->use_local_certs<br>";
echo "IPN-1d2-> In local Cert if clause, and not sure why here!
<br>";
            curl_setopt($ch, CURLOPT_CAINFO, "/home1/davelkwd/
wpappsforthat.com/wp-includes/certificates/ca-bundle.crt");
        }

        curl_setopt($ch, CURLOPT_FORBID_REUSE, 1);
        curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 30);
        curl_setopt($ch, CURLOPT_HTTPHEADER, array(
            'User-Agent: PHP-IPN-Verification-Script',
            'Connection: Close',
        ));

        $res = curl_exec($ch);

echo "IPN-1e-> res = $res<br>";
        // 09-27-22 might want to re-write this if clause as can
be misleading.
        if ( ! ($res)) {
            $errno = curl_errno($ch);
            $errstr = curl_error($ch);
            curl_close($ch);
            throw new Exception("cURL error: [$errno] $errstr");
        }

        $info = curl_getinfo($ch);
        $http_code = $info['http_code'];

        if ($http_code != 200) {
            throw new Exception("PayPal responded with http code 
$http_code");
        }

        curl_close($ch);

        // 09-06-22: LOCAL TEST HAS NO PP RETURNS
        // Check if PayPal verifies the IPN data, and if so, 
return true.

        if ($res == VALID) {
            $i = 0;
            $data_text=$txn_id=$txn_type = "";

            foreach ($_POST as $key => $value) {
                $i++;
                $data_text .= $key . "=" . $value . "\r\n";
                // Retrieve specific $key & $value
                if ($key == "txn_id") {
                    $txn_id = $value;
                }
                if ($key == "txn_type") {
                    $txn_type = $value;
                }
            }

            /* CUSTOM LOGFILE CREATION & WRITING TO IT - found 
alternate */

            date_default_timezone_set("America/Chicago");
            list($year, $month, $day, $hour, $minute, $second, 
$timezone) = explode(":", date("Y:m:d:H:i:s:T"));
            $timestamp = date_i18n('Ymd-His'); // Grab Time

            $write_to_db    = 1;
            $save_log_file  = 1;  // Set this to true to save a 
log file:
            $log_file_dir   = "/home1/davelkwd/
wpappsforthat.com/wp-content/uploads/paypal-ipn-logs/";  // 
08-22-22 took (partially) from product.php log directory ($a)
```

```
            $logfile        = $timestamp . ":" . "id-$txn_id" .
":". "txntyp-$txn_type" . ".txt";
            $filename       = "$log_file_dir" . "$logfile";

            // WRITE TO TEXT FILE (temporary process) // //

            if (($save_log_file == 1) && ($data_text)) {
                $fh = fopen($filename, 'w');
                fwrite($fh, $data_text);    //write data
                fclose($fh);                //close file
            }

            // ADD DATABASE SAVING HERE! //
                /*
                * DB codes for types of transactions
                *   payment_status=Completed => New record
                *   payer_status=verified
                *   payment_status=
                */


            if ($write_to_db) {
                // PENDING
            }

            // ************************* //

            //return true;
            return array (1,$logfile,$data_text);
        } else {
            // echo "IPN-Return below res=$res<br>";    // No
content for SIMULATOR VERSION
            return false;
        }
    }
}      // END ALL CLASS CODE //
```