

```

// API Setup Parameters
$gatewayURL = 'https://secure.payscout.com/api/v2/three-step';
$APIKey = '2F822Rw39fx762MaV7Yy86jXGTC7sCDy';

// If there is no POST data or a token-id, print the initial Customer
Information form to get ready for Step One.
if (empty($_POST['DO_STEP_1']) && empty($_GET['token-id'])) {

    print ' <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">';
    print '
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Collect non-sensitive Customer Vault Info </title>
  </head>
  <body>
    <p><h2>Step One: Collect non-sensitive Customer Vault information.<br
/></h2></p>

    <h3> Customer Information</h3>
    <h4> Billing Details</h4>

    <form action="" method="post">
      <table>
        <tr><td>Customer Vault Id </td><td><input type="text"
name="customer-vault-id" value=""></td></tr>
        <tr><td>Company</td><td><input type="text" name="billing-address-
company" value="Acme, Inc."></td></tr>
        <tr><td>First Name </td><td><input type="text" name="billing-
address-first-name" value="John"></td></tr>
        <tr><td>Last Name </td><td><input type="text" name="billing-
address-last-name" value="Smith"></td></tr>
        <tr><td>Address </td><td><input type="text" name="billing-address-
address1" value="1234 Main St."></td></tr>
        <tr><td>City </td><td><input type="text" name="billing-address-
city" value="Beverly Hills"></td></tr>
        <tr><td>State/Province </td><td><input type="text" name="billing-
address-state" value="CA"></td></tr>
        <tr><td>Zip/Postal </td><td><input type="text" name="billing-
address-zip" value="90210"></td></tr>
        <tr><td>Country </td><td><input type="text" name="billing-address-
country" value="US"></td></tr>
        <tr><td>Phone Number </td><td><input type="text" name="billing-
address-phone" value="555-555-5555"></td></tr>
        <tr><td>Email Address </td><td><input type="text" name="billing-
address-email" value="test@example.com"></td></tr>

        <tr><td><h4><br /> Shipping Details</h4>
        <tr><td>First Name </td><td><input type="text" name="shipping-
address-first-name" value="Mary"></td></tr>
        <tr><td>Last Name </td><td><input type="text" name="shipping-
address-last-name" value="Smith"></td></tr>
        <tr><td>Address </td><td><input type="text" name="shipping-address-
address1" value="1234 Main St."></td></tr>

```

```

        <tr><td>Suite</td><td><input type="text" name="shipping-address-
address2" value="Unit #2"></td></tr>
        <tr><td>City </td><td><input type="text" name="shipping-address-
city" value="Beverly Hills"></td></tr>
        <tr><td>State/Province </td><td><input type="text" name="shipping-
address-state" value="CA"></td></tr>
        <tr><td>Zip/Postal </td><td><input type="text" name="shipping-
address-zip" value="90210"></td></tr>
        <tr><td>Country</td><td><input type="text" name="shipping-address-
country" value="US"></td></tr>
        <tr><td colspan="2"> </td>
        <tr><td colspan="2" align=center><input type="submit" value="Submit
Step One"><input type="hidden" name="DO_STEP_1" value="true"></td></tr>
    </table>

    </form>
</body>
</html>

```

```

';
}else if (!empty($_POST['DO_STEP_1'])) {

    // Initiate Step One: Now that we've collected the non-sensitive customer
information, we can combine other customer information and build the XML
format.
    $xmlRequest = new DOMDocument('1.0','UTF-8');

    $xmlRequest->formatOutput = true;
    $xmlSale = $xmlRequest->createElement('add-customer');

    // Authentication, Redirect-URL are typically the bare minimum.
    appendXmlNode($xmlRequest, $xmlSale, 'api-key', $APIKey);
    appendXmlNode($xmlRequest, $xmlSale, 'redirect-
url', $_SERVER['HTTP_REFERER']);

    // Some additional fields may have been previously decided by user

    appendXmlNode($xmlRequest, $xmlSale, 'merchant-defined-field-1' , 'Red');
    appendXmlNode($xmlRequest, $xmlSale, 'merchant-defined-field-2',
'Medium');

    if(!empty($_POST['customer-vault-id'])) {
        appendXmlNode($xmlRequest, $xmlSale, 'customer-vault-id' ,
$_POST['customer-vault-id']);
    }

    // Set the Billing & Shipping from what was collected on initial shopping
cart form
    $xmlBillingAddress = $xmlRequest->createElement('billing');
    appendXmlNode($xmlRequest, $xmlBillingAddress, 'first-name',
$_POST['billing-address-first-name']);
    appendXmlNode($xmlRequest, $xmlBillingAddress, 'last-name',
$_POST['billing-address-last-name']);

```

```

        appendXmlNode($xmlRequest, $xmlBillingAddress, 'address1',
$_POST['billing-address-address1']);
        appendXmlNode($xmlRequest, $xmlBillingAddress, 'city', $_POST['billing-
address-city']);
        appendXmlNode($xmlRequest, $xmlBillingAddress, 'state', $_POST['billing-
address-state']);
        appendXmlNode($xmlRequest, $xmlBillingAddress, 'postal', $_POST['billing-
address-zip']);
        //billing-address-email
        appendXmlNode($xmlRequest, $xmlBillingAddress, 'country', $_POST['billing-
address-country']);
        appendXmlNode($xmlRequest, $xmlBillingAddress, 'email', $_POST['billing-
address-email']);
        appendXmlNode($xmlRequest, $xmlBillingAddress, 'phone', $_POST['billing-
address-phone']);
        appendXmlNode($xmlRequest, $xmlBillingAddress, 'company', $_POST['billing-
address-company']);
        $xmlSale->appendChild($xmlBillingAddress);

        $xmlShippingAddress = $xmlRequest->createElement('shipping');
        appendXmlNode($xmlRequest, $xmlShippingAddress, 'first-name',
$_POST['shipping-address-first-name']);
        appendXmlNode($xmlRequest, $xmlShippingAddress, 'last-name',
$_POST['shipping-address-last-name']);
        appendXmlNode($xmlRequest, $xmlShippingAddress, 'address1',
$_POST['shipping-address-address1']);
        appendXmlNode($xmlRequest, $xmlShippingAddress, 'city', $_POST['shipping-
address-city']);
        appendXmlNode($xmlRequest, $xmlShippingAddress, 'state', $_POST['shipping-
address-state']);
        appendXmlNode($xmlRequest, $xmlShippingAddress, 'postal',
$_POST['shipping-address-zip']);
        appendXmlNode($xmlRequest, $xmlShippingAddress, 'country',
$_POST['shipping-address-country']);
        appendXmlNode($xmlRequest, $xmlShippingAddress, 'address2',
$_POST['shipping-address-address2']);
        $xmlSale->appendChild($xmlShippingAddress);

$xmlRequest->appendChild($xmlSale);

// Process Step One: Submit all customer details to the Payment Gateway
except the customer's sensitive payment information.
// The Payment Gateway will return a variable form-url.
$data = sendXMLviaCurl($xmlRequest, $gatewayURL);

// Parse Step One's XML response
$gwResponse = @new SimpleXMLElement($data);
if ((string)$gwResponse->result == 1 ) {
    // The form url for used in Step Two below
    $formURL = $gwResponse->{'form-url'};
} else {
    throw New Exception(print " Error, received " . $data);
}

```

```

    // Initiate Step Two: Create an HTML form that collects the customer's
sensitive payment information
    // and use the form-url that the Payment Gateway returns as the submit
action in that form.
    print ' <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">';

    print '

        <html>
        <head>
            <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8" />
            <title>Collect sensitive Customer Info </title>
        </head>
        <body>';
    // Uncomment the line below if you would like to print Step One's
response
    // print '<pre>' . (htmlentities($data)) . '</pre>';

    print '
        <p><h2>Step Two: Collect sensitive payment information and POST
directly to payment gateway<br /></h2></p>

        <form action="'. $formURL. '" method="POST">
        <h3> Payment Information</h3>
        <table>
            <tr><td>Credit Card Number</td><td><INPUT type ="text"
name="billing-cc-number" value="4111111111111111"> </td></tr>
            <tr><td>Expiration Date</td><td><INPUT type ="text"
name="billing-cc-exp" value="1014"> </td></tr>
            <tr><td colspan="2" align=center><INPUT type ="submit"
value="Submit Step Two"></td> </tr>
        </table>
        </form>
        </body>
        </html>
        ';
    // NOTE: CVV cannot be stored, per PCI Requirements

} elseif (!empty($_GET['token-id'])) {

    // Step Three: Once the browser has been redirected, we can obtain the
token-id and complete
    // the Customer Vault Add through another XML HTTPS POST including the
token-id which abstracts the
    // sensitive payment information that was previously collected by the
Payment Gateway.
    $tokenId = $_GET['token-id'];
    $xmlRequest = new DOMDocument('1.0','UTF-8');
    $xmlRequest->formatOutput = true;
    $xmlCompleteTransaction = $xmlRequest->createElement('complete-action');
    appendXmlNode($xmlRequest, $xmlCompleteTransaction, 'api-key', $APIKey);
    appendXmlNode($xmlRequest, $xmlCompleteTransaction, 'token-id', $tokenId);
    $xmlRequest->appendChild($xmlCompleteTransaction);

```

```

// Process Step Three
$data = sendXMLviaCurl($xmlRequest,$gatewayURL);

$gwResponse = @new SimpleXMLElement((string)$data);
print ' <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">';
print '
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Step Three - Complete Add Customer </title>
</head>
<body>';

print "
  <p><h2>Step Three: Script automatically completes the process of
adding the Customer Vault record <br /></h2></p>";

if ((string)$gwResponse->result == 1 ) {
  print " <p><h3> Customer Vault was Added, XML response
was:</h3></p>\n";
  print '<pre>' . (htmlentities($data)) . '</pre>';

} elseif((string)$gwResponse->result == 2) {
  print " <p><h3> Customer Vault was Not Added</h3>\n";
  print " Reason : " . (string)$gwResponse->{'result-text'} ." </p>";
  print " <p><h3>XML response was:</h3></p>\n";
  print '<pre>' . (htmlentities($data)) . '</pre>';
} else {
  print " <p><h3> Customer Vault Add caused an Error.</h3>\n";
  print " Error Description: " . (string)$gwResponse->{'result-text'}
." </p>";
  print " <p><h3>XML response was:</h3></p>\n";
  print '<pre>' . (htmlentities($data)) . '</pre>';
}
print "</body></html>";

} else {
print "ERROR IN SCRIPT<BR>";
}

function sendXMLviaCurl($xmlRequest,$gatewayURL) {
// helper function demonstrating how to send the xml with curl

$ch = curl_init(); // Initialize curl handle
curl_setopt($ch, CURLOPT_URL, $gatewayURL); // Set POST URL

$headers = array();
$headers[] = "Content-type: text/xml";

```

```

    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers); // Add http headers to
let it know we're sending XML
$xmlString = $xmlRequest->saveXML();
    curl_setopt($ch, CURLOPT_FAILONERROR, 1); // Fail on errors
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // Allow redirects
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Return into a variable
    curl_setopt($ch, CURLOPT_PORT, 443); // Set the port number
    curl_setopt($ch, CURLOPT_TIMEOUT, 15); // Times out after 15s
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xmlString); // Add XML directly in
POST

    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);

    // This should be unset in production use. With it on, it forces the ssl
cert to be valid
    // before sending info.
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);

    if (!( $data = curl_exec($ch) ) ) {
        print "curl error =>" . curl_error($ch) . "\n";
        throw New Exception(" CURL ERROR :" . curl_error($ch));
    }
    curl_close($ch);

    return $data;
}

// Helper function to make building xml dom easier
function appendXmlNode($domDocument, $parentNode, $name, $value) {
    $childNodes = $domDocument->createElement($name);
    $childNodesValue = $domDocument->createTextNode($value);
    $childNodes->appendChild($childNodesValue);
    $parentNode->appendChild($childNodes);
}

```