

VevoPay

Version 1.0.0

VevoPay Integration Guide

VevoCart.com

support@vevocart.com

www.vevocart.com

Table of Contents

1.	Introduction	2
2.	VevoPay Overview.....	2
3.	Payment Gateway Types	3
3.1	Hosted Payment Gateways.....	3
3.2	Integrated Payment Gateways (Seamless Checkout)	4
3.3	Offline Credit Card Payment	6
4.	Integrating with Hosted Payment Gateways	7
4.1	Hosted Payment Gateways Transaction Flow	7
4.2	Posting-to-VevoPay Web Page	9
4.3	HostedRequest XML Data Descriptions	11
4.3.1	< Authentication > Element.....	11
4.3.2	<ShoppingCart> Element	12
4.3.3	<HostedInformation> Element	13
4.3.4	HostedRequest Example	15
5.	Integrating with Integrated Payment Gateways (Seamless Checkout)	16
5.1	Integrated Payment Gateway Transaction Flows	16
5.2	Offline Credit Card Payment Transaction Flows.....	18
5.3	Credit Card Input Page Descriptions	20
5.4	Payment Token Descriptions	24
5.5	ChargeRequest XML Data Descriptions.....	25
5.5.1	< Authentication > Element.....	25
5.5.2	<ChargeInformation> Element.....	25
5.5.3	< RecurringInformation > Element	26
5.5.4	ChargeRequest Example.....	27
5.6	ChargeResponse XML Data Descriptions	28
6.	Generating VevoPaySignature.....	31
7.	Authentication XML Data Description.....	32

1. Introduction

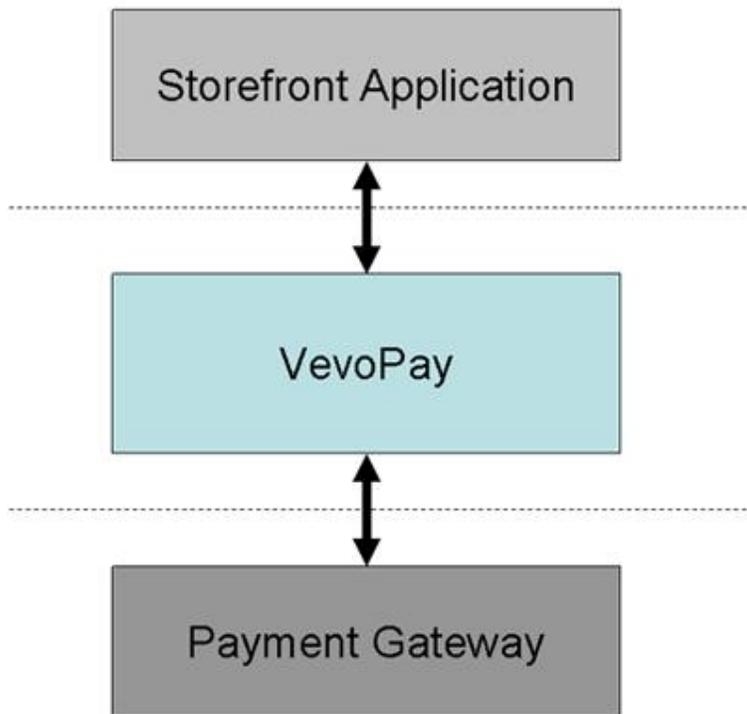
This document was written to explain the communication model of VevoPay and the data descriptions. It is good for developers who want to integrate VevoPay with storefront applications.

The document is divided to three main sections. Overview section explains general characteristic description of VevoPay. Transaction flow section explains communication process between VevoPay and storefront application. And the data description section explains technical data details.

2. VevoPay Overview

VevoPay is an ASP.NET PCI compliant payment application which is designed and implemented to meet all PCI compliance requirements. Currently, VevoPay has already been fully audited by the qualified assessor and is in the final stage of being PA-DSS officially certified. There is no source code change expected between the current version and officially certified version.

VevoPay can operate with various storefront application platform (ASP.NET, ASP, PHP, etc.) and is integrated with world famous gateways (for example: PayPal, 2Checkout, RBS WorldPay, etc.). It is designed to run independently as standalone application.



VevoPay simplifies the process to be a PCI compliant store. Developers can integrate the storefront application with VevoPay and let VevoPay to handle the payment process. There is no need to migrate your store to a completely new storefront application to become a PCI compliant store.

Only the highlighted portion of data communication, which is between your storefront application and VevoPay, will need to be added by a developer.

VevoPay is also distributed with VevoCart application, serving as a payment module. Still, it can work with other storefront applications with some customization/integration efforts.

To integrate storefront application to work with VevoPay, you should review transaction flows, communication methods, and configuration settings in this document.

3. Payment Gateway Types

To begin integration with VevoPay, you first need to know which payment gateways to use. Then, depending on the type of your payment gateway, the communication with VevoPay will occur at the different steps of your checkout process.

There are three main types of payment gateways that VevoPay handles: Hosted Integrated, and Offline.

Hosted Payment Gateway – Examples of this type are PayPal Standard and 2Checkout. Visitors of your store website will be redirected to the payment gateway website during the checkout to enter payment and credit card information.

Integrated Payment Gateway (Seamless Checkout) – Examples of this type are PayPal Website Payments Pro US, eWay. Visitors will enter the payment information including credit card information directly into the website.

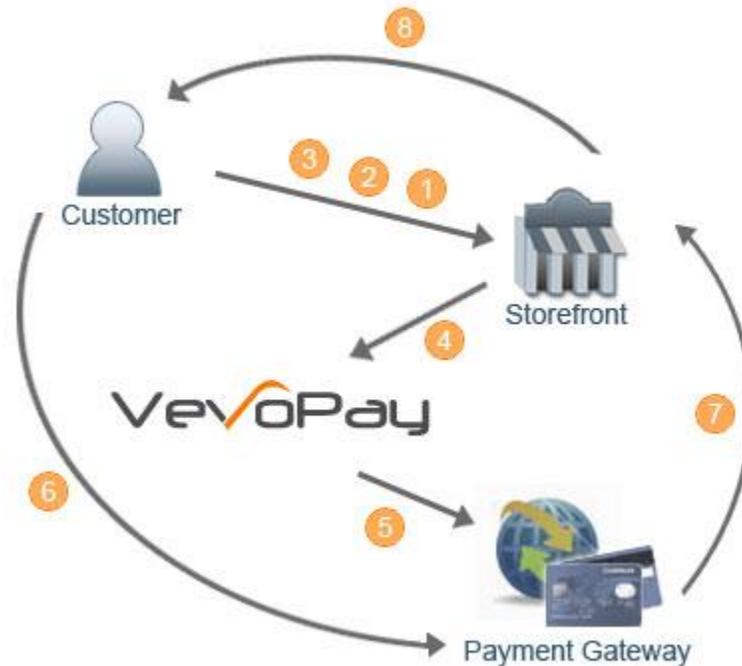
Offline Credit Card Payment – This type payment will store credit card information in the database to be used later by merchant. It does not contact the online payment gateway automatically.

3.1 Hosted Payment Gateways

For Hosted Payment Gateway, VevoPay only needs to know when your customer has finished all the checkout steps and is ready to place the order. The Hosted Payment Gateways supported in VevoPay are:

- 2Checkout
- PayPal Website Payments Standard
- RBS WorldPay

The regular flow for checkout will be similar to below.



- 1) The customer has added items to the shopping cart.
- 2) The customer starts the checkout process on the store website.
- 3) The customer enters all necessary information such as shipping address, shipping method, coupon, etc. Payment information is not entered yet.
- 4) **Once the customer confirms the order, the store website will send data to VevoPay to start the payment process.**
- 5) VevoPay will redirect the browser to the payment gateway's website, such as PayPal website.
- 6) The customer enters payment information on the payment gateway's website.
- 7) **The payment gateway's website will send payment result back to the original store website whether the payment has completed or not.**
- 8) Storefront shows order complete page to customer.

The store website will need to handle items in **bold**, which are #4 and #7, to communicate with VevoPay and payment gateway's website. These should be the only modifications by developers to integrate VevoPay with Hosted Payment Gateways.

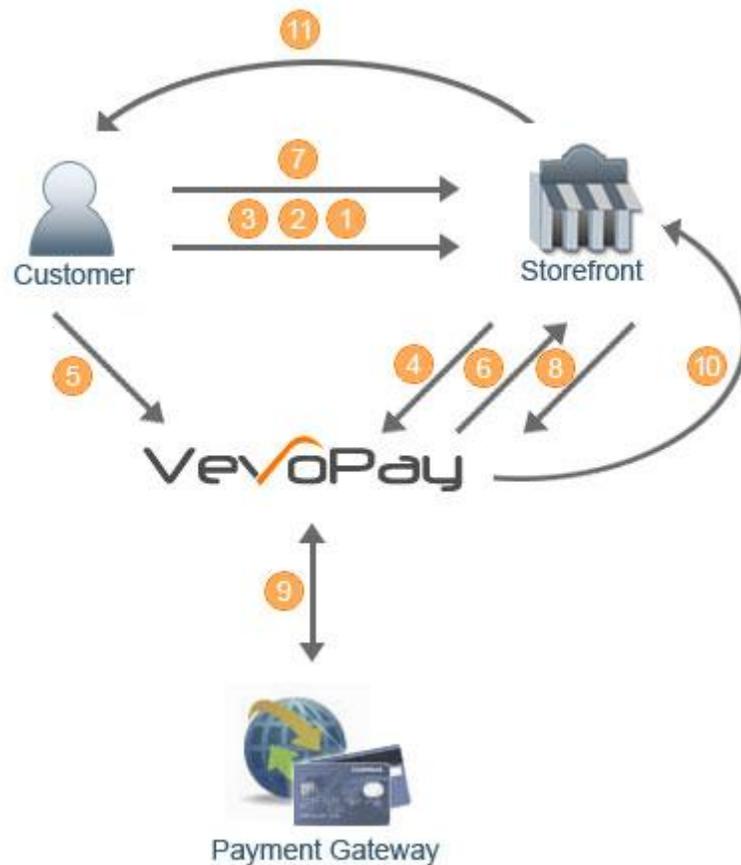
Note: Result data that are returned in step #7 vary for each payment gateway. It needs to learn specific format by reading Payment Gateway's integration guide.

3.2 Integrated Payment Gateways (Seamless Checkout)

For Integrated Payment Gateways, Customer needs to enter credit card information into VevoPay. The credit card data will then be forwarded to the payment gateway website for actual processing. VevoPay supports the following Integrated Payment Gateways:

- Authorize.Net
- eWay
- LinkPoint
- Moneris
- PayPal Payflow Pro
- PayPal Website Payments Pro US
- PayPal Website Payments Pro UK
- ProtX
- QuickBooks Merchant Service

Their checkout flows are different from Hosted Payment Gateway's flows.



- 1) The customer has added items to the shopping cart.
- 2) The customer starts the checkout process on the store website.
- 3) The customer enters all necessary information such as shipping address, shipping method, coupon, etc.
- 4) **In the credit card data entry page, the store website will open VevoPay's CreditCardInfo.aspx page (usually in an <iframe> HTML tag).**

- 5) The customer enters payment information into VevoPay directly, whose data entry page resides in a separated iframe.
- 6) **VevoPay will send payment token data back and redirect to the store website.**
- 7) The customer confirms payment process.
- 8) **Storefront sends payment charge request to VevoPay.**
- 9) VevoPay sends payment information to payment gateway website and receives payment result.
- 10) **VevoPay will send payment result back to the storefront.**
- 11) Storefront shows order complete page to customer.

The store website will need to handle items in **bold**, which are #4, #6, #8 and #10, to communicate with VevoPay and payment gateway's website. These should be the only modifications by developers to integrate VevoPay with Integrated Payment Gateway.

3.3 Offline Credit Card Payment

Offline Credit Card Payments are similar to Integrate Payment Gateway, but they do not have to interact with online payment gateway websites. The credit card data will be encrypted in VevoPay's database and can be viewed later via VevoPay's web interface. There is only one payment method for this payment category:

- Offline Credit Card Payment (internal to VevoPay)

The regular flow for checkout will be similar to below:



- 1) The customer has added items to the shopping cart.
- 2) The customer starts the checkout process on the store website.
- 3) The customer enters all necessary information such as shipping address, shipping method, coupon, etc.

- 4) **In the credit card data entry page, the store website will open VevoPay's CreditCardInfo.aspx page (usually in an <iframe> HTML tag).**
- 5) The customer enters payment information into VevoPay directly, whose data entry page resides in a separated iframe.
- 6) **VevoPay will send payment token data back and redirect to the store website.**
- 7) The customer confirms payment process.
- 8) **Storefront sends payment charge request to VevoPay.**
- 9) VevoPay encrypts and saves credit card information in its database. There is no interaction with online payment gateways for the Offline Credit Card Payment.
- 10) **VevoPay will send payment result back to the storefront.**
- 11) Storefront shows order complete page to customer.

The store website will only need to handle items in **bold**, which are #4, #6, #8 and #10, to communicate with VevoPay. These should be the only modifications by developers to integrate VevoPay with Integrated Payment Gateway.

The data format will be described in more details in the "Transaction Flow" and "Data Description" chapters of this document below.

4. Integrating with Hosted Payment Gateways

This chapter describes general steps to use Hosted Payment Gateways in VevoPay. We assume that you already have a storefront website with a complete checkout process.

The discussion will focus on the modifications that a developer will need to make to integrate an existing storefront website to use VevoPay's payment feature.

4.1 Hosted Payment Gateways Transaction Flow

Below is the transaction flow for Hosted Payment Gateways. After customers have finished general checkout steps and are ready to complete the order, they will be sent to the payment gateway's website (e.g. PayPal) to enter payment information (e.g. credit card) there.

The items in **bold**, which are #4, #6, #13, and #16, may need to be added or modified by the developers to communicate with VevoPay and Hosted Payment Gateways.

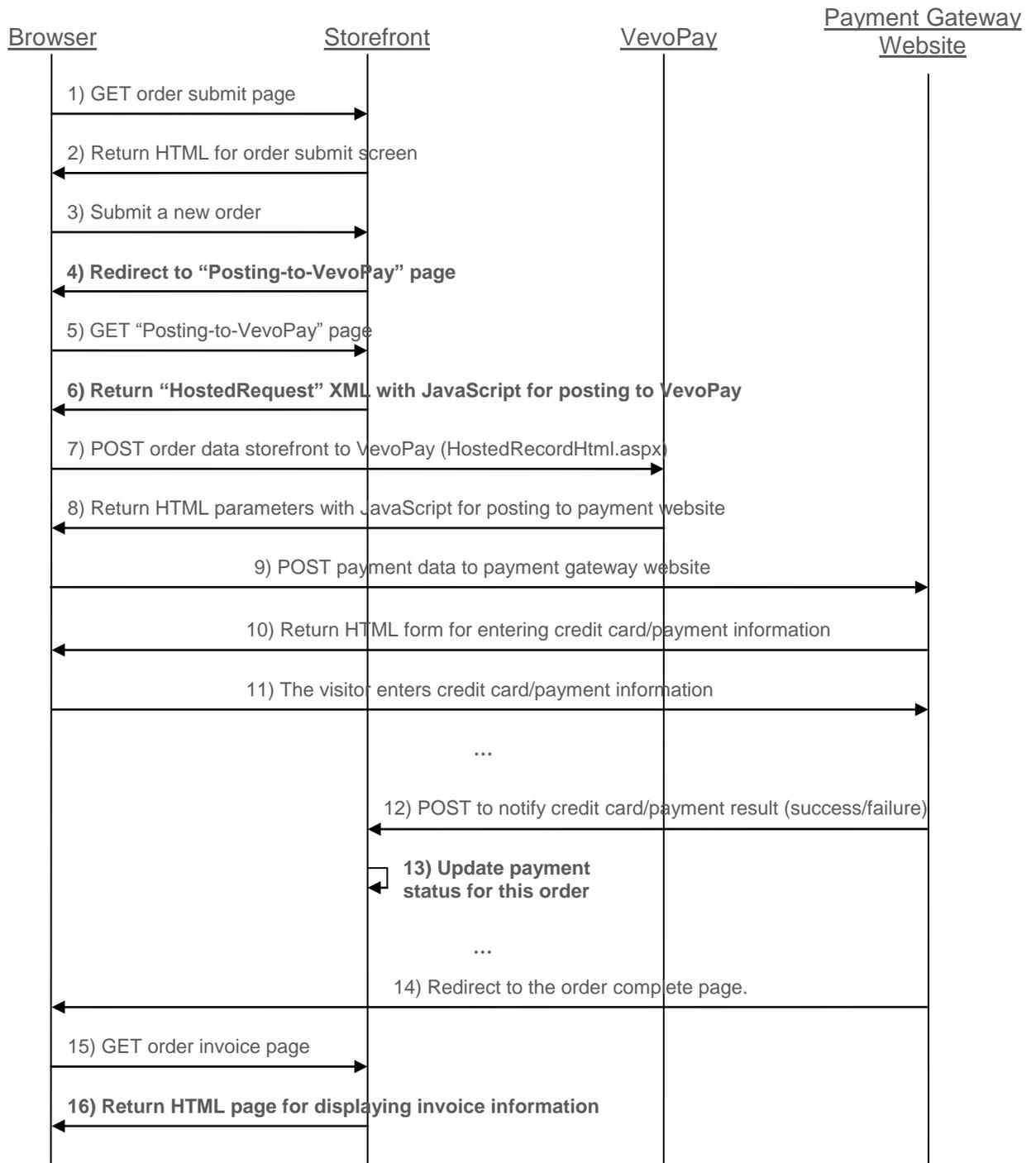


Figure 1: Dataflow for Hosted Payment Gateways

The diagram above shows the flow after a visitor has started the checkout process and about to view the order submit page.

- 1) The browser is about to visit the order submit page.
- 2) The browser displays order confirmation screen from storefront application.
- 3) The visitor submits the order to storefront application.
- 4) **Storefront returns a HTTP redirection command to the “Posting-to-VevoPay” page.**
- 5) The browser requests the storefront’s “Posting-to-VevoPay” page.
- 6) **The storefront return “Posting-to-VevoPay” page. This page contains “HostedRequest” XML parameters and a JavaScript that will post order data (e.g. amount) automatically to VevoPay.**
- 7) The browser displays the page from step 6). The “HostedRequest” XML will be posted to the “HostedRecordHtml.aspx” page on VevoPay website by the JavaScript.
- 8) VevoPay processes the incoming “HostedRequest” XML parameters and re-format the data to meet payment gateway’s expectation. The newly-formatted data will be returned on this VevoPay’s web page (HostedRecordHtml.aspx). This page also contains a JavaScript that will post order data automatically to the payment gateway website (e.g. paypal.com).
- 9) The browser displays data from VevoPay and post the data to payment gateway website by the JavaScript.
- 10) The payment gateway website (e.g. paypal.com) returns HTML form for entering credit card/payment information.
- 11) The visitor enters credit card and/or other payment information to complete the payment.
- 12) After the payment gateway website has processed the data entered by the visitor, it returns the payment result by posting to storefront.
- 13) **The storefront processes payment result from the payment gateway website (e.g. PayPal’s IPN). Depending on your storefront, this may include update the order’s payment status in the database.**
- 14) Once the visitor has completed, the payment gateway website will send a HTTP redirection to browse back to the storefront website.
- 15) The browser requests the order invoice page (e.g. www.yourdomain.com/OrderInvoice.aspx).
- 16) **The storefront returns HTML for the order invoice page.**

4.2 Posting-to-VevoPay Web Page

The “Posting-to-VevoPay” web page here is referring to item 6) in the Hosted transaction flow diagram. The developer needs to add a new page in the storefront website (e.g. www.yourdomain.com/VevoPayPosting.aspx).

This web page will include “HostedRequest” parameters to be posted to VevoPay, which will be described in more details in the next section. The “HostedRequest” XML will be inside the hidden input tag named “HostedXml” as shown below.

Note: The XML will need to be encoded to conform to HTML standard (e.g. “<” will be encoded to “<”).

```
<input name="HostedXml" type="hidden" value=" [Encoded HostedRequest XML]" />
```

The web page will need to include a self-posting JavaScript that will post to the URL of the “HostedRecordHtml.aspx” page in the VevoPay website (e.g. <https://www.yourdomain.com/VevoPay/HostedRecordHtml.aspx>). For example:

```
<script language="javascript" type="text/javascript">
    document.forms[0].action = '[URL of the VevoPay's
    HostedRecordHtml.aspx page]';
    document.forms[0].__VIEWSTATE.value = '';
    document.forms[0].__VIEWSTATE.name = 'NOVIEWSTATE';
    document.forms[0].submit();
</script>
```

An example of a complete HTML for Posting-to-VevoPay would be below:

```
<body>
    <form name="aspnetForm" method="post"
    action="GatewayPosting.aspx?OrderID=1" id="aspnetForm">
    <div>
    <input type="hidden" name="HostedXml" id="HostedXml"
    value=" %3c%3fxml+version%3d%221.0%22%3f%3e%3cHostedRequest%3e+++3cA
    uthentication%3e+++++3cItem%3e+++++3cName%3ePaymentByPayPal
    Email%3c%2fName%3e+++++3cValue%3eVevo%40systems.com%3c%2fValue
    %3e+++++3c%2fItem%3e+++++3cItem%3e+++++3cName%3ePaymentB
    yPayPalEnvironment%3c%2fName%3e+++++3cValue%3eTrue%3c%2fValue%
    3e+++++3c%2fItem%3e+++3c%2fAuthentication%3e+++3cShoppingCart%3e
    +++3cCartItem%3e+++++3cProductID%3e1%3c%2fProductID%3e+
    ++++3cQuantity%3e+1%3c%2fQuantity%3e+++++3cNam
    e%3e+Test+Product%3c%2fName%3e+++++3cPrice%3e+29.0000%3c%
    2fPrice%3e+++++3cTangible%3e+True%3c%2fTangible%3e+++++
    +++3c%2fCartItem%3e+++++3c%2fShoppingCart%3e+++3cHostedInformatio
    n%3e+++++3cBillingAddress%3e+++++3cFirstName%3etest%3c%2fF
    irstName%3e+++++3cLastName%3etest%3c%2fLastName%3e+++++
    %3cCompany%3e%3c%2fCompany%3e+++++3cAddress1%3e982932329%3c%2
    fAddress1%3e+++++3cAddress2%3e%3c%2fAddress2%3e+++++3c
    City%3eBKK%3c%2fCity%3e+++++3cState%3e%3c%2fState%3e+++++
    ++3cZip%3e20293%3c%2fZip%3e+++++3cCountry%3eTH%3c%2fCountry%
    3e+++++3cPhone%3e%3c%2fPhone%3e+++++3cFax%3e%3c%2fFax%
    3e+++++3c%2fBillingAddress%3e+++++3cShippingAddress%3e+++++
    +++3cFirstName%3etest%3c%2fFirstName%3e+++++3cLastName%3etes
    t%3c%2fLastName%3e+++++3cCompany%3e%3c%2fCompany%3e+++++
    +3cAddress1%3e982932329%3c%2fAddress1%3e+++++3cAddress2%3e%3
    c%2fAddress2%3e+++++3cCity%3eBKK%3c%2fCity%3e+++++3cSt
    ate%3e%3c%2fState%3e+++++3cZip%3e20293%3c%2fZip%3e+++++
    %3cCountry%3eTH%3c%2fCountry%3e+++++3cPhone%3e%3c%2fPhone%3e+
    ++++3cFax%3e%3c%2fFax%3e+++++3c%2fShippingAddress%3e+++++
    +3cEmail%3etest%40127.0.01%3c%2fEmail%3e+++++3cPaymentName%3ePay
    Pal%3c%2fPaymentName%3e+++++3cAmount%3e49.0000%3c%2fAmount%3e++++
    +++3cInvoiceNumber%3e%3c%2fInvoiceNumber%3e+++++3cCurrencyCode%
    3eAUD%3c%2fCurrencyCode%3e+++++3cItemName%3ePayment+for+Store++++
```

```

+++%3c%2fItemName%3e+++++++%3cIPAddress%3e127.0.0.1%3c%2fIPAddress%3
e+++++++%3cVerifyCode%3e%3c%2fVerifyCode%3e+++++++%3cReturnUrl%3eht
p%3a%2f%2fyourdomain%2fStore%2fGateway%2fPayPalReturn.aspx%3fOrderID
%3d3%3c%2fReturnUrl%3e+++++++%3cCancelUrl%3ehttp%3a%2f%2fyourdomain%
2fStore%2f%3c%2fCancelUrl%3e+++++++%3cNotifyUrl%3ehttp%3a%2f%2fyourd
omain%2fStore%2fGateway%2fPayPalNotification.aspx%3c%2fNotifyUrl%3e+
++++%3cVevoPaymentSignature%3eu6EOkP7yR6TK2e1%2fPzxh2%2f10wfw1taEJ
51MLgMOJjUvVF%2bQBP0W%2fQxzntn721VQUm2YtIRtIuOBbRHb7J%2bJ9aw0Ms3tS7k
fo+++++++%3c%2fVevoPaymentSignature%3e+++%3c%2fHostedInformation%3e%
3c%2fHostedRequest%3e" />

```

```

<h1>Processing Your Order...</h1>
<p>Please do NOT hit 'BACK' or 'REFRESH' button</p>
<p>Please wait while we are transferring you to the payment
information page.</p>
<p></p>
<script language="javascript" type="text/javascript">
    document.forms[0].action =
    "http://www.yourdomain.com/vevopay/HostedRecordHtml.aspx";
    document.forms[0].__VIEWSTATE.value = '';
    document.forms[0].__VIEWSTATE.name = 'NOVIEWSTATE';
    document.forms[0].submit();
</script>
</div>

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

```

4.3 HostedRequest XML Data Descriptions

This XML will be posted to VevoPay during the payment step in Transaction Flow item 6). The XML root element is “<HostedRequest>” element, which will enclose the other child elements as shown below:

```

<HostedRequest>
  <Authentication>
    ...
  </Authentication>
  <ShoppingCart>
    ...
  </ShoppingCart>
  <HostedInformation>
    ...
  </HostedInformation>
</HostedRequest>

```

4.3.1 < Authentication > Element

This is a common element that is used by several XML commands. Please see chapter 6 for more detailed information for the <Authentication> element.

4.3.2 <ShoppingCart> Element

The <ShoppingCart> tag provides shopping cart. It has the following child nodes:

```

<ShoppingCart>
  <CartItem>
    <ProductID> ProductID1 </ProductID>
    <Quantity> Quantity1 </Quantity>
    <Name> + Name1 + </Name>
    <Price> + Price1 + </Price>
    <Tangible> + Tangible1 + </Tangible>
  </CartItem>
  <CartItem>
    <ProductID> ProductID2 </ProductID>
    <Quantity> Quantity2 </Quantity>
    <Name> + Name2 + </Name>
    <Price> + Price2 + </Price>
    <Tangible> + Tangible2 + </Tangible>
  </CartItem>
  .
  .
  .
  <CartItem>
    <ProductID> ProductID X </ProductID>
    <Quantity> Quantity X </Quantity>
    <Name> + Name X + </Name>
    <Price> + Price X + </Price>
    <Tangible> + Tangible X + </Tangible>
  </CartItem>
</ShoppingCart>

```

PARAMETER	DESCRIPTION	EXAMPLE
ProductID	Your product ID	10
Quantity	Order quantity for each product	2
Name	Your Product Name	Football kit
Price	Your Product Price per item	100
Tangible	Flag whether product is e-good or not. "False" for e-good product and "True" for not e-good product.	True

4.3.3 <HostedInformation> Element

The <HostedInformation> element provides data about billing address, shipping address and other information for Hosted Payment Gateway. It consists of the following child nodes:

```

<HostedInformation>
  <BillingAddress>
    <FirstName> First Name </FirstName>
    <LastName> Last Name </LastName>
    <Company> Company </Company>
    <Address1> Address1 </Address1>
    <Address2> Address2 </Address2>
    <City> City </City>
    <State> State </State>
    <Zip> Zip </Zip>
    <Country> Country </Country>
    <Phone> Phone </Phone>
    <Fax> Fax </Fax>
  </BillingAddress>
  <ShippingAddress>
    <FirstName> First Name </FirstName>
    <LastName> Last Name </LastName>
    <Company> Company </Company>
    <Address1> Address1 </Address1>
    <Address2> Address2 </Address2>
    <City> City </City>
    <State> State </State>
    <Zip> Zip </Zip>
    <Country> Country </Country>
    <Phone> Phone </Phone>
    <Fax> Fax </Fax>
  </ShippingAddress>
  <Email> Email </Email>
  <PaymentName> PaymentName </PaymentName>
  <Amount> Amount </Amount>
  <InvoiceNumber> OrderID </InvoiceNumber>
  <CurrencyCode> Currency Code </CurrencyCode>
  <ItemName> SiteName </ItemName>
  <IPAddress> IP Address </IPAddress>
  <ReturnUrl> Return URL </ReturnUrl>
  <CancelUrl> Cancel URL </CancelUrl>
  <NotifyUrl> Notify URL </NotifyUrl>
  <VevoPaymentSignature> + paymentSignature.CreateKey( "", Amount,
orderID ) + </VevoPaymentSignature>
</HostedInformation>
    
```

PARAMETER	DESCRIPTION	EXAMPLE
FirstName	Customer first name.	David
LastName	Customer last name.	Dunn
Company	Customer company.	VevoCart
Address1	Customer address1.	777 abc road
Address2	Customer address2.	Khlongsan
City	Customer city.	Bangkok
State	Customer state.	LA
Zip	Customer zip.	10600
Country	Customer country.	TH
Phone	Customer phone number.	023456789
Fax	Customer fax number.	023456790
Email	Customer email.	abc@mail.com
PaymentName	Payment Method Name	PayPal
Amount	Order total	200
InvoiceNumber	Order ID	17
CurrencyCode	Your Currency code	USD
ItemName	Your website name	VevoCart
IPAddress	Customer IP address	127.0.0.1
ReturnUrl	The URL to show to the customer if the payment has gone through completely.	www.yourdomain.com/Gateway/PayPalReturn.aspx
CancelUrl	The URL to show to the customer if the payment has been canceled.	www.yourdomain.com
NotifyUrl	The URL to receive the notification from the online payment gateway websites. This is the web page that the developer has to implemented in item 13) in the Transaction Flow in section 4.1 above.	www.yourdomain.com/gateway/paypalnotification.aspx

VevoPaymentSignature	Unique string that sends to VevoPay to validate the request for security reason. This is explained in chapter 6, "Generating VevoPay Signature".	IB9ZySv8QD5Uo6l38 PoEo6Xd0sxezntXy CJE3v8HCfmgfYub
----------------------	---	--

4.3.4 HostedRequest Example

The XML below shows an example of the complete "HostedRequest" XML.

```

<?xml version="1.0"?>
<HostedRequest>
  <Authentication>
    <Item>
      <Name>PaymentByPayPalEmail</Name>
      <Value>premier_ian@gmail.com</Value>
    </Item>
    <Item>
      <Name>PaymentByPayPalEnvironment</Name>
      <Value>True</Value>
    </Item>
  </Authentication>
  <ShoppingCart>
    <CartItem>
      <ProductID>66</ProductID>
      <Quantity> 1</Quantity>
      <Name> Adidas Polo Shirt</Name>
      <Price> 33.75</Price>
      <Tangible> True</Tangible>
    </CartItem>
  </ShoppingCart>
  <HostedInformation>
    <BillingAddress>
      <FirstName>User</FirstName>
      <LastName>One</LastName>
      <Company></Company>
      <Address1>11</Address1>
      <Address2></Address2>
      <City>LA</City>
      <State>CA</State>
      <Zip>90007</Zip>
      <Country>US</Country>
      <Phone></Phone>
      <Fax></Fax>
    </BillingAddress>
    <ShippingAddress>
      <FirstName>User</FirstName>
      <LastName>One</LastName>
      <Company></Company>
      <Address1>11</Address1>
  </HostedInformation>
</HostedRequest>
    
```

```

        <Address2></Address2>
        <City>LA</City>
        <State>CA</State>
        <Zip>90007</Zip>
        <Country>US</Country>
        <Phone></Phone>
        <Fax></Fax>
    </ShippingAddress>
    <Email>user1@127.0.0.1</Email>
    <PaymentName>PayPal</PaymentName>
    <Amount>39.75</Amount>
    <InvoiceNumber>17</InvoiceNumber>
    <CurrencyCode>USD</CurrencyCode>
    <ItemName>Payment for VevoCart           </ItemName>
    <IPAddress>127.0.0.1</IPAddress>
    <VerifyCode></VerifyCode>
    <ReturnUrl>http://localhost/WebApp/Gateway/PayPalReturn.aspx?OrderID=17</ReturnUrl>
    <CancelUrl>http://localhost/WebApp/</CancelUrl>
    <NotifyUrl>http://localhost/WebApp/Gateway/PayPalNotification.aspx</NotifyUrl>
    <VevoPaymentSignature>TetpzkwH6lsugMUMqiyCtur+4OZzzRrdquLboS4fBmj8uz/gOSDcmazYYXTDgRB6A+5DFCcxVVcYP/LsaOpCKfA==</VevoPaymentSignature>
    </HostedInformation>
</HostedRequest>

```

Please remember that this XML will be inside another HTML tag:

```

<input name="HostedXml" type="hidden" value=" [Encoded HostedRequest XML]">.

```

Therefore, the XML below text should be encoded to HTML format accordingly to be put inside the <input> tag, such as encoding "<" to "<,"

5. Integrating with Integrated Payment Gateways (Seamless Checkout)

5.1 Integrated Payment Gateway Transaction Flows

Below is the transaction flow for Integrated Payment Gateways. They require credit card information to be entered in VevoPay’s website. Then, these credit card data will be sent out to the payment gateway websites (e.g. Authorize.NET) for the actual processing.

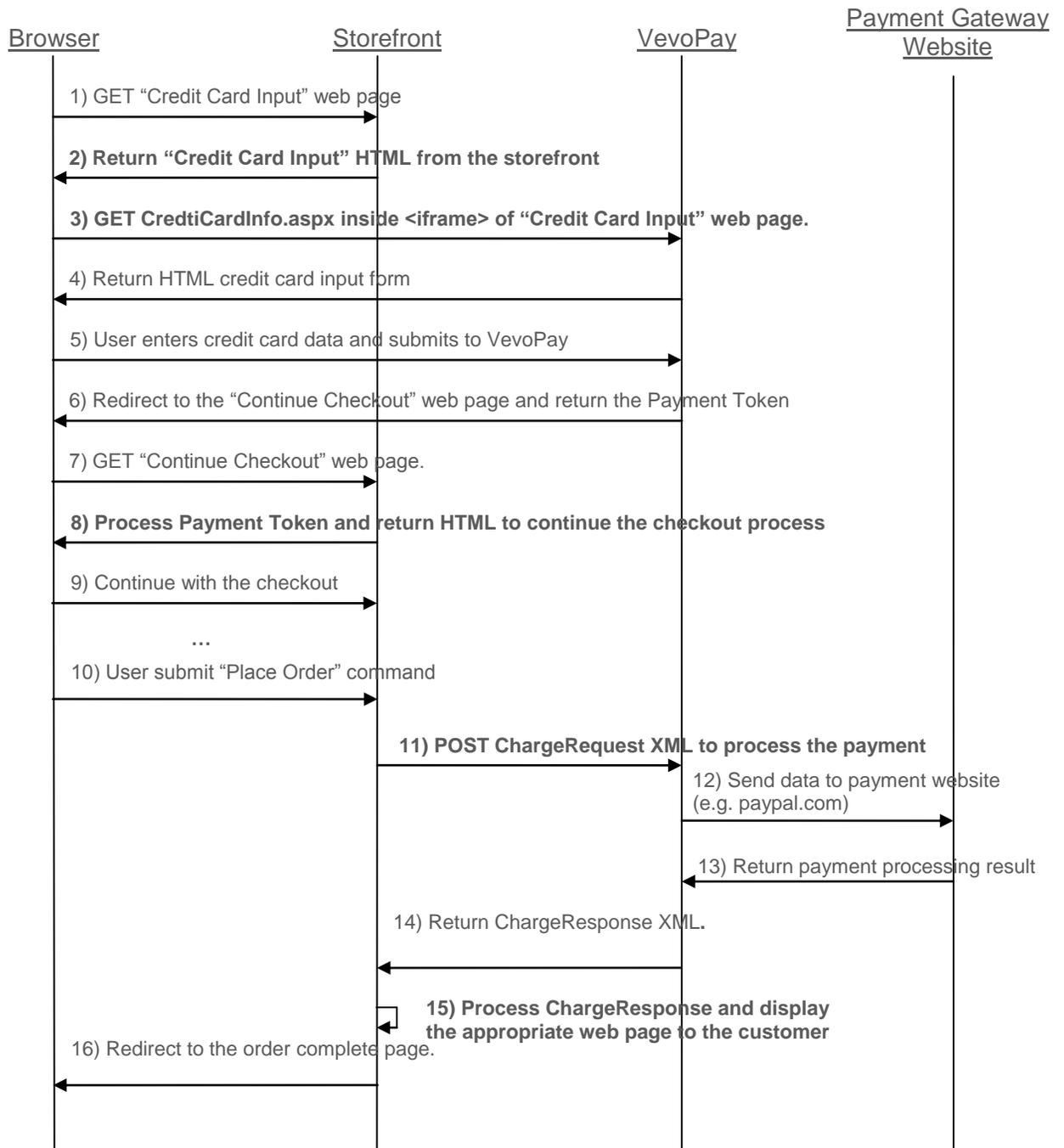


Figure 3: Dataflow for Integrated Payment Gateways

The diagram above shows the flow after a visitor has started the checkout process and about to view the credit card input form.



- 1) The browser visits the “Credit Card Input” web page from the storefront.
- 2) **The storefront returns “Credit Card Input” HTML web page data to the browser. This page includes <iframe> tag to display VevoPay’s CreditCardInfo.aspx.**
- 3) **Because of the <iframe> tag, the browser requests VevoPay’s CreditCardInfo.aspx page (e.g. <https://www.yourdomain.com/VevoPay/CreditCardInfo.aspx>). This page contains the actual form to receive credit card data on VevoPay.**
- 4) The VevoPay HTML data is received on the browser.
- 5) The visitor has entered the credit card data and submitted to VevoPay.
- 6) VevoPay keeps this data for later use and return a HTTP redirect command with Payment Token to the “Continue Checkout” web page in the storefront.
- 7) The browser requests “Continue Checkout” web page in the storefront.
- 8) **The storefront processes the Payment Token in the query string and return HTML to continue the checkout process.**
- 9) The visitor browses on storefront website to continue with the normal checkout.
- 10) In the last checkout page, the visitor clicks “Place Order”, which will also charge credit card.
- 11) **Payment data, including credit card data, will be put in “ChargeRequest” XML formatted by storefront. The XML will be sent out to VevoPay.**
- 12) VevoPay formats the input data and send them to payment gateway websites (e.g. paypal.com) to process the actual charge.
- 13) The payment website returns the payment result to VevoPay.
- 14) VevoPay returns the “ChargeResponse” XML to the storefront.
- 15) **The storefront processes the “ChargeResponse” XML (e.g. updating payment status in the database).**
- 16) Storefront returns a HTTP response to show the checkout complete page.

5.2 Offline Credit Card Payment Transaction Flows

Below is the transaction flow for Offline Credit Card Payments. It will store credit card information in the VevoPay to be processed later. It does not contact the online payment gateway. Merchant needs to process payment transaction and update order status manually.

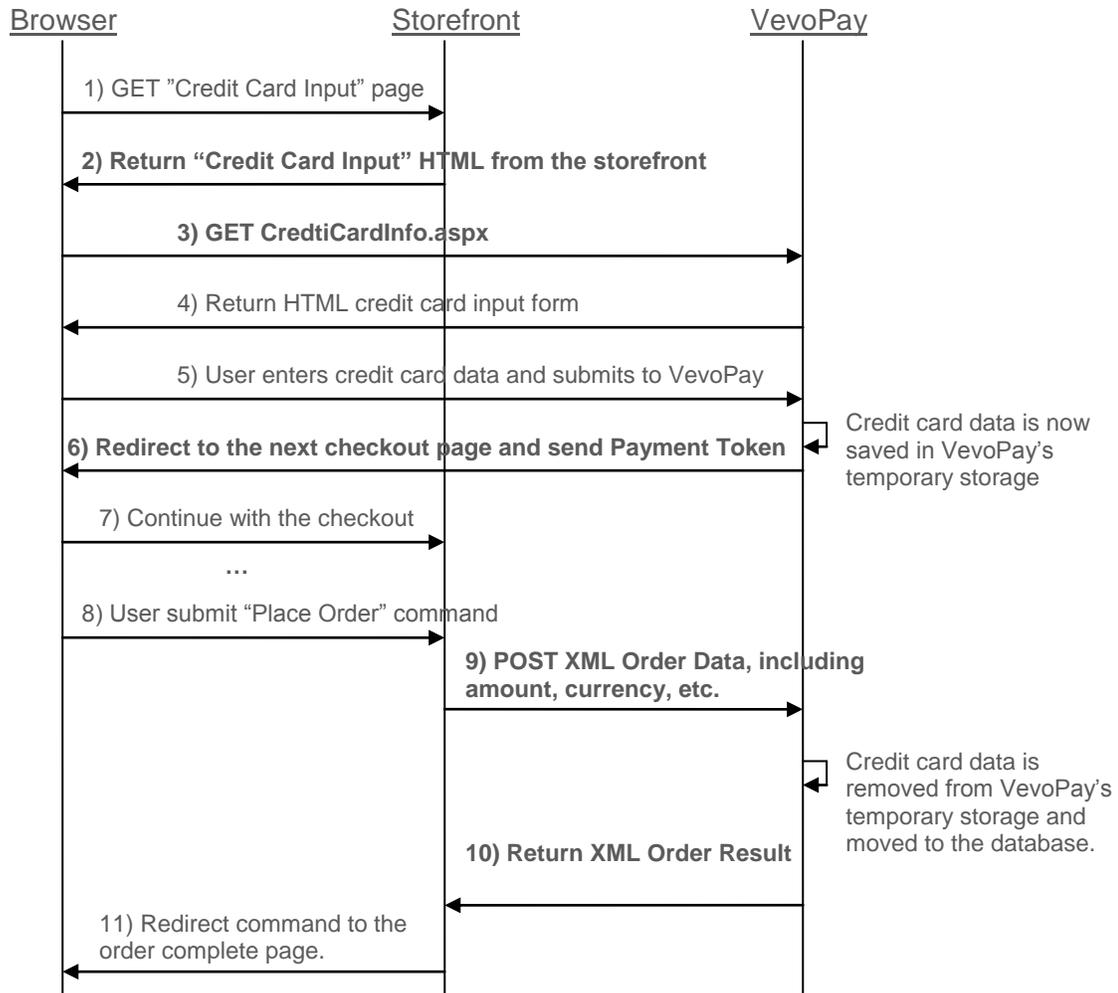


Figure 3: Dataflow for Offline Credit Card Payment

The diagram above shows the flow after a visitor has started the checkout process and about to view the credit card input form.

- 1) **The browser is about to visit the Credit Card Input page.**
- 2) The HTML data is received on the browser.
- 3) **The browser requests VevoPay’s CreditCardInfo.aspx page (e.g. <https://www.yourdomain.com/VevoPay/CreditCardInfo.aspx>). This page has the actual forms to receive credit card data.**
- 4) The HTML data is received on the browser.
- 5) The visitor has entered the credit card data and submitted to VevoPay.
- 6) **VevoPay keeps this data for later use and return a HTTP redirect command with Payment Token to continue the checkout process.**

- 7) The visitor browses on storefront website to continue with the normal checkout.
- 8) In the last checkout page, the visitor clicks "Place Order", which will also charge credit card.
- 9) **Payment data, including credit card data, will be put in XML format by storefront. The XML will be sent out to VevoPay. VevoPay will then move credit card data from the temporary storage to its database.**
- 10) **VevoPay returns the XML Order Result to storefront.**
- 11) Storefront receives the payment result and returns a HTTP redirection to order complete page.

5.3 Credit Card Input Page Descriptions

To keep your website look-and-feel, you may use "iframe" HTML tag in your application to link to "CreditCardInfo.aspx" in VevoPay, which is the page where the customer enters credit card information.

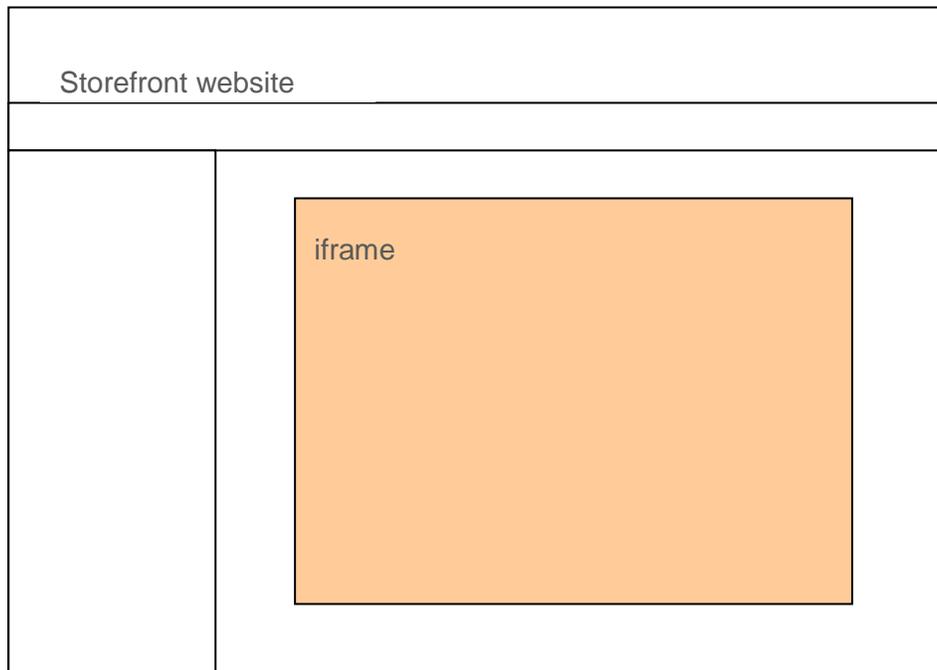


Figure 4: Example of IFrame layout for storefront website.

You may put iframe code into the payment page such as the example below:

```
<iframe id="uxPaymentFrame" width="478px" height="800px"
frameborder="0">
...
...
...
</iframe>
```

```
...
</iframe>
```

The CreditCardInfo.aspx in VevoPay receives information from storefront application via query strings in the URL. For example, the URL inside iframe tag to show the credit card input page maybe:

```
http://www.yourdomain.com/VevoPay/CreditCardInfo.aspx?Cvv2Required=True&BillingAddressRequired=True&SupportedCreditCards=Visa%2cMasterCard%2cDiscover%2cAmerican+Express&SupportedCreditCardValues=Visa%2cMasterCard%2cDiscover%2cAmex&BillingFirstName=John&BillingLastName=Smith&BillingCompany=Smith+Company&BillingAddress1=21+West+52nd+Street&BillingAddress2=&BillingCity=New+York&BillingZip=10021&CurrentCountry=United+States&CurrentState=New+York&BillingPhone=12345678&BillingFax=&Email=johmsmith%40mail.com&ShippingFirstName=John&ShippingLastName=Smith&ShippingCompany=Smith+Company&ShippingAddress1=21+West+52nd+Street&ShippingAddress2=&ShippingCity=New+York&ShippingZip=10021&ShippingCountry=United+States&ShippingState=New+York&ShippingPhone=12345678&ShippingFax=&IPAddress=169.2.2.13&skiplogin=False&IsOfflineData=False&StoreTheme=Default&a1=Credit&a2=Type&a21=Please+Select&a3=Credit+Number&a4=Cardholder+Name&a5=Credit+Verify&a6=Expiration&a7=Card+Issue&a8=CardStart&b1=Billing+Address&b2=Firstname&b3=Lastname&b4=Country&b5=State&b6=Company&b7=Address&b8=City&b9=Zip&RedirectURL=htp%3a%2f%2fwww.yourdomain.com%2fstore%2fCheckoutPaymentInfo.aspx
```

The following are full description of required parameters for CreditCardInfo.aspx.

PARAMETER	DESCRIPTION	EXAMPLE
Cvv2Required	Flag whether this payment method requires CVV2 to be entered by the customers.	True
BillingAddressRequired	Flag whether this payment method requires billing address.	True
SupportedCreditCards	The list of supported credit card of this payment method. Separated by commas. One example is "Visa, MasterCard". This is the list of credit cards that will be shown in the credit card drop-down when the customer enters credit card information in the Credit Card Input page. This list should have the same order of credit cards as SupportedCreditCardValues.	Visa, MasterCard, American Express

SupportedCreditCardValues	<p>The value list of supported credit card of this payment method. It separated by commas. One example is "Visa, MasterCard" to support Visa and Master Card.</p> <p>The Values in this field may be different for each payment gateway requirements. For example, MasterCard in ProtX use "MC" but most other gateways use "MasterCard".</p>	Visa, MasterCard, Amex
BillingFirstName	Billing address's first name.	David
BillingLastName	Billing address's last name.	Dunn
BillingCompany	Billing address's company.	VevoCart
BillingAddress1	Billing address's address1.	777 abc road
BillingAddress2 (optional)	Billing address's address2.	Khlongsan
BillingCity	Billing address's city.	Bangkok
BillingZip	Billing address's zip.	10600
CurrentCountry	Billing address's country.	TH
CurrentState	Billing address's state.	LA
BillingPhone	Billing address's telephone number.	023456789
BillingFax	Billing address's fax number.	023456790
Email	Billing address's email address.	abc@mail.com
ShippingFirstName	Shipping address's first name.	David
ShippingLastName	Shipping address's last name.	Dunn
ShippingCompany	Shipping address's company.	VevoCart
ShippingAddress1	Shipping address's address1.	777 abc road
ShippingAddress2	Shipping address's address2.	Khlongsan
ShippingCity	Shipping address's city.	Bangkok
ShippingZip	Shipping address's zip.	10600

ShippingCountry	Shipping address's country.	TH
ShippingState	Shipping address's state.	LA
ShippingPhone	Shipping address's telephone number.	023456789
ShippingFax	Shipping address's fax number.	023456790
IPAddress	Customer IP address	127.0.0.1
skiplogin	Flag whether this payment process required login before checkout.	False
IsOfflineData	Flag whether this payment using offline credit card payment method.	True
StoreTheme	Get your style sheet theme	Default
a1	"Credit Card" text label.	Credit Card
a2	"Credit Card Type" text label.	Credit Card Type
a21	"Please Select" text in credit card type selected drop down.	-- Please select --
a3	"Credit Card Number" text label.	Credit Card Number
a4	"Card Holder Name" text label.	Card Holder Name
a5	"CVV" text label.	CVV
a6	"Expiration date" text label	Expiration Date
a7	"Card Issue" text label	Card Issue
a8	"Card start" text label	Card Start
b1	"Billing Address" text label	Billing Address
b2	"First Name" text label	First Name
b3	"Last Name" text label	Last Name
b4	"Country" text label	Country
b5	"State" text label	State
b6	"Company" text label	Company
b7	"Address" text label	Address

b8	“City” text label	City
b9	“Zip” text label	Zip
RedirectURL	Page’s URL of storefront what is redirected from VevoPay and receives Payment Token after customer inputs credit card information.	http://www.yourdomain.com/CheckoutPaymentInfo.aspx

5.4 Payment Token Descriptions

After customer inputs credit card information, Payment Token data is returned to storefront application via the query string on redirected page. This data is used in item 8) of Integrated Payment Gateway (5.1) and Offline Credit card payment (5.2) chapters. Payment token data consists of:

PARAMETER	DESCRIPTION	EXAMPLE
Token	Payment Token number	21394892
FirstName	Customer’s first name	David
LastName	Customer’s last name	Dunn
Company	Customer’s company name	VevoCart
Address1	Customer’s address 1	777 abc road
Address2	Customer’s address 2	Khlongsan
City	Customer’s city name	Bangkok
Zip	Customer’s zip code	10600
State	Customer’s state	LA
Country	Customer’s country	TH
Phone	Customer’s phone number	023456789

Once the credit card information is entered, VevoPay will send an HTTP redirection command. The URL in the redirection command would be something similar to below:

```
http://www.yourdomain.com/orderconfirm.aspx?Token=21394892&FirstName=David&LastName=Dunn&Company=VevoCart&Address1=777 abc
```

```
road&Address2= Khlongsan&City=Bangkok&Zip=10600
&State=LA&Country=TH&Phone=023456789
```

5.5 ChargeRequest XML Data Descriptions

Once the order placement is confirmed by customers at storefront application, the application must POST XML data to VevoPay for payment processing.

This XML will be posted to VevoPay during the payment step in 9) of Integrated Payment Gateway Transaction Flow and Offline Credit Card Payment. The XML root element is “<ChargeRequest>” element, which will enclose the other child elements as shown below:

```
<ChargeRequest>
  <Authentication>
    ...
  </Authentication>
  <<ChargeInformation>
    ...
  </ChargeInformation>
  <RecurringInformation>
    ...
  </RecurringInformation>
</ChargeRequest>
```

5.5.1 < Authentication > Element

This is a common element that is used by several XML commands. Please see chapter 6 for more detailed information for the <Authentication> element.

5.5.2 <ChargeInformation> Element

The < ChargeInformation > tag provides charge information data. It has the following child nodes:

```
<ChargeInformation>
  <PaymentName> PaymentName </PaymentName>
  <Token> Token </Token>
  <IPAddress> IPAddress </IPAddress>
  <Amount> Amount </Amount>
  <CurrencyCode> CurrencyCode </CurrencyCode>
<SiteName> SiteName </SiteName>
  <OrderID> OrderID </OrderID>
  <NotifyUrl> NotifyUrl </NotifyUrl>
  <IsLastProduct> IsLastProduct </IsLastProduct>
  <VevoPaymentSignature> +
paymentSignature.CreateKey( Token, Amount,  orderID ) +
</VevoPaymentSignature>
</ChargeInformation>
```

PARAMETER	DESCRIPTION	EXAMPLE
PaymentName	Payment Method Name	PayPal Pro
Token	This is the unique token id that returns from CreditCardInfo.aspx page. For get the correct customer details when place order.	5c46d670-6b96-4934-9794-1799c1cda344
IPAddress	Customer IP address	127.0.0.1
Amount	Order total	200
CurrencyCode	Your Currency Code	USD
SiteName	Your website name	VevoCart
NotifyUrl	Your URL when payment gateways notify the payment result. (use for some gateway)	www.yourdomain.com/gateway/paypalnotification.aspx
IsLastProduct	<p>Flag whether it's the last product of this transaction.</p> <p>This parameter use with recurring order, which is more that one product in shopping cart.</p> <p>The VevoPay will process payment one by one until the "IsLastProduct" value set to "True".</p> <p>For the non-recurring order should also set to "True".</p>	True
VevoPaymentSignature	Unique key that send to VevoPay to verify correct order.	IB9ZySv8QD5Uo6l38PoEo6Xd0sxezntXyCJE3v8HCfmgfYub

5.5.3 < RecurringInformation > Element

It is used for recurring payment. It consists of the following child notes:

```

<RecurringInformation>
  <ProductName> ProductName </ProductName>
  <RecurringInterval> RecurringInterval </RecurringInterval>
  <RecurringIntervalUnit>RecurringIntervalUnit
    </RecurringIntervalUnit>
  <RecurringNumberOfTrialCycles>RecurringNumberOfTrialCycles
    </RecurringNumberOfTrialCycles>
  <RecurringNumberOfRegularCycles>RecurringNumberOfRegularCycles
    </RecurringNumberOfRegularCycles>
  <RecurringRegularTotal>RecurringRegularTotal
    </RecurringRegularTotal>
  <RecurringTrialTotal>RecurringTrialTotal</RecurringTrialTotal>
</RecurringInformation>

```

PARAMETER	DESCRIPTION	EXAMPLE
PaymentName	Payment Method Name	
RecurringInterval	Recurring time for 1 cycle	1
RecurringIntervalUnit	Recurring unit for 1 cycle. Can be Day, Week, Month, or Year	Month
RecurringNumberOfTrialCycles	Number of trial recurring cycles. This value can be zero.	2
RecurringNumberOfRegularCycles	Number of regular recurring cycles.	4
RecurringRegularTotal	Regular amount per cycle.	200
RecurringTrialTotal	Trial amount per cycle. This value can be zero.	100

5.5.4 ChargeRequest Example

Below is an example of the complete ChargeRequest XML to be sent to VevoPay.

```

<?xml version="1.0"?>
<ChargeRequest>
  <Authentication>
    <Item>
      <Name>PayPalAPIUserName</Name>
      <Value>nuudee_apil.sanook.com</Value>
    </Item>
    <Item>
      <Name>PayPalAPIPassword</Name>
      <Value>Z7JNHEZ9YD32AKNC</Value>
    </Item>
  </Authentication>
</ChargeRequest>

```

```

        </Item>
        <Item>
            <Name>PayPalAPISignature</Name>
            <Value>AFcWxV21C7fd0v3bYYYRCpSSRl31Ab4Ta-
ufHdCvyADP5S9QseAYqjAa</Value>
        </Item>
        <Item>
            <Name>PayPalEnvironment</Name>
            <Value>sandbox</Value>
        </Item>
        <Item>
            <Name>UseCustomTimeZone</Name>
            <Value>False</Value>
        </Item>
        <Item>
            <Name>CustomTimeZone</Name>
            <Value>-12.00</Value>
        </Item>
    </Authentication>
    <ChargeInformation>
        <PaymentName>PayPal Pro</PaymentName>
        <Token>5c46d670-6b96-4934-9794-1799c1cda344</Token>
        <IPAddress>127.0.0.1</IPAddress>
        <Amount>0</Amount>
        <CurrencyCode>USD</CurrencyCode>
        <SiteName>VevoCart</SiteName>
        <OrderID></OrderID>

        <NotifyUrl>http://localhost/WebApp/Gateway/PayPalNotification.aspx</NotifyUrl>
        <IsLastProduct>True</IsLastProduct>
    <VevoPaymentSignature>
r0R2Q50uJ6MG/1CbJBrmx4ctxNuGNmwerwweIxRjUY0K0+VAwlibNcIXvQxF4xsT/xXT
4MRDL7P96ROvc8farNguuMzVYAtyJ9M9GdaJY5lOd2FPwhgpS/nZfYRm3VP5
    </VevoPaymentSignature>
    </ChargeInformation>
    <RecurringInformation>
        <ProductName>Honda City</ProductName>
        <RecurringInterval>5</RecurringInterval>
        <RecurringIntervalUnit>Day</RecurringIntervalUnit>

        <RecurringNumberOfTrialCycles>1</RecurringNumberOfTrialCycles>
        <RecurringNumberOfRegularCycles>3</RecurringNumberOfRegularCycles>
    >
        <RecurringRegularTotal>20000</RecurringRegularTotal>
        <RecurringTrialTotal>17</RecurringTrialTotal>
    </RecurringInformation>
</ChargeRequest>
    
```

5.6 ChargeResponse XML Data Descriptions

For the Integrated Payment Gateway, it will return response data to VevoPay after finishing payment process. Then VevoPay will generate data in xml format and return to storefront application. Following XML is the example which is returned from VevoPay:

```
<?xml version="1.0" encoding="utf-16"?>
<ChargeResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Status>OK</Status>
  <GatewayOrderID> 200 <GatewayOrderID />
  <ErrorMessage />
  <PaymentLog>TIMESTAMP=2010%2d08%2d02T10%3a18%3a57Z&amp;CORRELATIONID
=69d1218fcd69&amp;ACK=Success&amp;VERSION=52%2e0&amp;BUILD=1421734&
&amp;AMT=23%2e00&amp;CURRENCYCODE=AUD&amp;AVSCODE=X&amp;CVV2MATCH=M&a
mp;TRANSACTIONID=7AR01800CX255714K</PaymentLog>
  <ReferenceID />
  <CvvStatus>Pass</CvvStatus>
  <AvsAddrStatus>Pass</AvsAddrStatus>
  <AvsZipStatus>Pass</AvsZipStatus>
</ChargeResponse>
```

This data is used in #12 of Integrated Payment Gateway and 10# Offline Credit card payment steps.

PARAMETER	DESCRIPTION	EXAMPLE
Status	Flag whether the payment process is complete or not. Pass: for payment process complete Fail: for payment process not complete	Pass
GatewayOrderID	The Gateway Order Number or Gateway Transaction number which is returned from payment gateway	200
ErrorMessage	The error message which is returned when payment status is "Fail"	Error (10527): Invalid Data This transaction cannot be processed. Please enter a valid credit card number and type.

PaymentLog	The raw data returned from payment gateway. It can be logged by the storefront for later troubleshooting.	TIMESTAMP=2010%2d08%2d02T10%3a18%3a57Z&CORRELATIONID=69d1218fcd69&ACK=Success&
ReferenceID	The reference ID which is returned when process payment with recurring product (for PayPal Pro US only)	
CvvStatus	<p>The CVV response status which is returned after process payment finished. The response status it can be:</p> <p>Pass: for the CVV which pass verification</p> <p>Fail: for the CVV which fail.</p> <p>Unavailable: for the CVV which do not return verification type or the credit card that is not supported.</p>	Pass
AvsAddrStatus	<p>The address verification status which is returned after process payment finished. The response status it can be:</p> <p>Pass: for the address which pass verification</p> <p>Fail: for the address which fail.</p> <p>Unavailable: for the address that not returns verification type or the Country which is not supported.</p>	Pass

AvsZipStatus	<p>The address verification status which is return after process payment finished. This status is returned from some payment gateway which separate verification between address and zip. The response status it can be:</p> <p>Pass: for the zip code which pass verification</p> <p>Fail: for the zip code which fail</p> <p>Unavailable: for the zip code which does not return verification type or the Country that not supported.</p>	Pass
--------------	--	------

6. Generating VevoPaySignature

From 4.3.3 and 5.5.2, <VevoPaymentSignature> element is included in XML Data because this information must be used for VevoPay license verification. It needs to be generated dynamically by using VevoPay Signature Library (VevoPaymentSignature.dll).

In case of Storefront website is developed by ASP.NET, you can use VevoPay Signature Library by adding VevoPaymentSignature.dll to project and import it to source code by:

```
using Vevo.PaymentSignatureLib;
```

And create PaymentSignatureServices object by:

```
PaymentSignatureServices paymentSignature = new  
PaymentSignatureServices();
```

Payment Signature must be created by CreateKey method of PaymentSignatureServices object. This description of method's parameters is

PARAMETER	DESCRIPTION	EXAMPLE
Token	This is the unique token id that returns from CreditCardInfo.aspx page. For get the correct customer details when place order.	5c46d670-6b96-4934-9794-1799c1cda344
Amount	Total amount of payment	200
OrderID	Order ID. It is identification number of order from storefront	20312

Below is a sample code for VevoPay Signature generating:

```
string VevoPaySignatureData =
    paymentSignature.CreateKey(paymentInfo.PaypalToken,
    paymentInfo.OrderTotal.ToString(), OrderGuid.ToString())
```

7. Authentication XML Data Description

This chapter describes the Authentication XML, which is a common component inside several other XML commands sending to VevoPay (e.g. HostedRequest, ChargeRequest).

The <Authentication> mainly specifies the account information (e.g. user name, password, or key) for the online payment gateways. The online payment gateways need this information to identify the merchant accounts and validate the requests to process the payments.

Under <Authentication> root node, there will be following data inside.

```
<Authentication>
  <Item>
    <Name> Name1 </Name>
    <Value> Value1 </Value>
  </Item>
  <Item>
    <Name> Name2 </Name>
    <Value> Value2 </Value>
  </Item>
  .
  .
  .
  <Item>
    <Name> Name X </Name>
    <Value> Value X </Value>
  </Item>
```

</Authentication>

PARAMETER	DESCRIPTION	EXAMPLE
Name	The Configuration Name	2COMerchantAccount
Value	The Configuration Value	Account1

Each <Item> element represents a name-value pair that specifies one payment parameter to VevoPay. Each payment gateway expects different name-value pairs

For example, 2Checkout payment requires “2COMerchantAccount” and “2COTestMode”. The storefront website needs to create <Authentication> element that contains two <item> tags. One <item> tag contains the <Name> element for “2COMerchantAccount” and “2COTestMode”.

The following table contains the list of required value for each payment gateway.

PARAMETER	DESCRIPTION	EXAMPLE
2Checkout		
2COMerchantAccount	Your 2CO merchant account	Account1
2COTestMode	Flag whether is test mode or not.	True
Authorize.Net		
AuthorizeTestMode	Flag whether is test mode or not.	True
AuthorizeLoginID	Your Authorize.Net API Login ID	58s2SHRdsEA
AuthorizeTransectionKey	Your Authorize.Net Transaction Key	22r2Y8UXw58ltZ8a
eWay		
eWayCustomerID	Your eWay account ID	98765432
eWayTest	Flag whether is test mode or not	True

eWayErrorCode	Set error code for testing transaction error. This setting for test mode only.	00
LinkPoint		
LinkPointStoreNumber	Your LinkPoint's Store ID	1921681123
LinkPointHost	Your LinkPoint's host name	staging.linkpt.net
LinkPointPort	Your LinkPoint's port	1129
LinkPointMode	<p>Set Your LinkPoint's mode</p> <ol style="list-style-type: none"> 1. "Live" for live environment mode. 2. "Good" for test approve response mode. 3. "Decline" for test decline response mode. 4. "Duplicate" for test duplicate response mode. 	Live
Moneris		
MonerisTestMode	Flag whether is test mode or not.	True
MonerisStoreID	Your Moneris's StoreID	Store1
MonerisAPI_Token	Your Moneris's API token	Afh632
MonerisUseCVD	Flag whether Use CVD in Moneris payment or not.	True

MonerisCrypt	Set Your Secure transaction "7" for SSL Enabled "8" for Non Secure Transaction	7
Payflow Pro		
PayflowProTest	Flag whether is test mode or not.	True
PayflowProUser	Your PayFlow's User Account	Vevo
PayflowProVendor	Your Payflow's Vendor name	Vevo
PayflowProPartner	Your Payflow's Partner	PayPal
PayflowProPassword	Your Payflow's Password	VevoPass
PayPal		
PayMentByPayPalEmail	Your PayPal's Email	vevo@systems.com
PaymentByPayPalEnvironment	Flag whether is test mode or not	True
PayPal Pro		
PayPalAPIUserName	Your PayPal's API User Name	Vevo_api1@systems
PayPalAPIPassword	Your PayPal's API Password	Y6IMGDY8ZE43BLPD
PayPalAPISignature	Your PayPal's API Signature	deNe2346Ndfsldfi\$jkN

PayPalEnvironment	Set your PayPal's Environment 1. "live" for use in live environment 2. "sandbox" for use in sandbox environment 3. "beta-sandbox" for use in beta-sandbox environment.	sandbox
UseCustomTimeZone	Flag whether to use your custom time zone or use PayPal time zone	True
CustomTimeZone	Set your custom time zone value.	-5.00
PayPal Pro UK		
PayPalProUKTest	Flag whether is test mode or not.	True
PayPalProUKUser	Your PayPalProUK's User Account	Vevo
PayPalProUKVender	Your PayPalProUK's Vendor name	Vevo
PayPalProUKPartner	Your PayPalProUK's Partner	PayPal
PayPalProUKPassword	Your PayPalProUK's Password	VevoPass

ProtX		
ProtXMode	Set your ProtX's mode 1. "LIVE" for live environment mode. 2. "TEST" for test mode. 3. "SIMULATOR" for simulator mode.	LIVE
ProtXVendorName	Your ProtX's vendor name	Vevovendor
QuickBooks QBMS		
QbmsAppID	Your Quickbooks's application ID.	152548313
QbmsAppLogin	You Quickbooks's application login	vevocart.qbms.com
QbmsConnectionTicket	You Quickbooks's connection ticket	QBM-54-2dthGweRE2s
QbmsTest	Flag whether is test mode or not	True
RBSWorldPay		
PaymentByRBSWorldPayInstId	Your WorldPay's installation ID	254789

<p>PaymentByRBSWorldPayMode</p>	<p>Set your WorldPay's mode</p> <ol style="list-style-type: none"> 1. "CAPTURED" test with result value captured. 2. "REFUSED" test with result value refused. 3. "AUTHORISED" test with result value authorized. 4. "ERROR" test with result value error. 5. "LIVE" for live environment mode 	<p>CAPTURED</p>
---------------------------------	---	-----------------