



Payment page integration in an Iframe

Contents

1. PRESENTATION.....	4
2. RESTRICTIONS.....	4
3. DECLARING AN IFRAME.....	4
4. DISPLAYING THE PAYMENT FORM IN THE IFRAME.....	5
5. ACTIVATING THE IFRAME MODE.....	5

6. HIDING THE 3D SECURE LOGOS.....	6
7. CUSTOMIZING INPUT FIELDS.....	6
8. TRANSMITTING THE RETURN URLS.....	6
9. TRANSMITTING THE TRANSACTION RECEIPT TO THE BUYER.....	8
10. SPECIFYING THE DISPLAY MODE OF THE RETURN PAGE.....	8
10.1. Displaying the return page in full screen.....	8
10.2. Displaying the return page in the parent frame.....	9
10.3. Displaying the return page in the same iframe.....	9
11. CATCHING THE END OF PAYMENT.....	10
12. OFFERING ADDITIONAL PAYMENT ATTEMPTS.....	10
13. MANAGING TIMEOUTS.....	10

1. PRESENTATION

The payment gateway allows you to use an iframe to embed the payment pages into your website.

This kind of integration allows the buyer to proceed to the payment without leaving the purchase tunnel.

This document presents the technical aspects of the iframe mode. It does not replace the Hosted Payment Page Implementation guide.



Using the iframe mode requires the merchant web site to use the HTTPS protocol on all of its pages.

2. RESTRICTIONS

Some payment methods are not compatible with integration by iframe. Here is a non-exhaustive list:

- 3x 4x Oney
- Giropay
- iDeal
- PayPal
- Sofort Banking

Iframe mode is not recommended if you propose payment methods using redirection to a third party website.



In order to ensure full compatibility with all the authentication servers, we do not recommend using the **SANDBOX** attribute (https://www.w3schools.com/tags/att_iframe_sandbox.asp).

3. DECLARING AN IFRAME

1. Create an HTML **<iframe>** tag on the page where you would like to display the payment page (this page should correspond to the "Payment" stage of your purchase journey).
2. Use the **name** attribute to give a name to your iframe. This name will be used in the payment form to specify within which item the form will be displayed.
3. Populate the **src** attribute with **your payment form URL**.
4. Define the size of the iframe by using the **width** and **height** attributes.
5. Use the **scrolling** attribute of the iframe tag to delete borders and scrollbars.

Example:

```
<iframe name="nameFrame" src="payment_form.php" width="600" height="550" scrolling="no" />
```

4. DISPLAYING THE PAYMENT FORM IN THE IFRAME

Modify your payment form by using the **target** attribute of your **<form>** tag.

Populate it with the name of your iframe.

```
<form method="POST" action="https://paiement.systempay.fr/vads-payment/" target="nameFrame" >
...
</form>
```

5. ACTIVATING THE IFRAME MODE

Populate **vads_action_mode** to **IFRAME** in your payment form in order to enable iframe mode and to obtain a clean display.

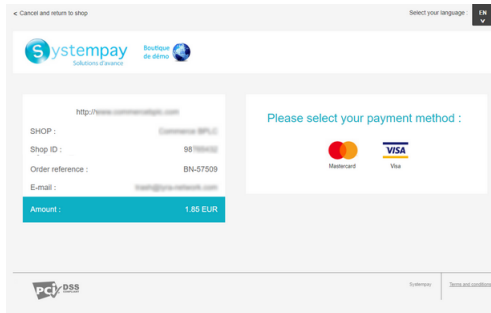


Figure 1: Redirect payment page without IFRAME

Please select your payment method :

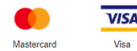


Figure 2: Redirect payment page with IFRAME



Using the payment page in IFRAME mode requires that the field **vads_action_mode** be valued at **IFRAME** and not to **INTERACTIVE**.

Example of integration:

1. Creating a HTML tag **<iframe>** on the page where you would like to display the fill out form (this page should correspond to the "Payment" stage of your purchase journey):

```
... <body> <iframe id="idFrame" name="nameFrame" src="https://www.mystore.com/payment/
form.php"/> </body> </html>
```

2. Value the attribute **src** with **your payment form URL**.

6. HIDING THE 3D SECURE LOGOS

Use the **3DS_LOGOS** attribute of **vads_theme_config** to hide the "Visa Secure" and "Mastercard Identity check" logos on the payment page.

```
vads_theme_config="3DS_LOGOS=false"
```

Figure 3: Payment page with 3D Secure logos

Figure 4: Payment page without 3D Secure logos

7. CUSTOMIZING INPUT FIELDS

Use the **vads_iframe_options** field to customize:

- the background color of input fields (thanks to the **fieldsBackgroundColor** attribute),
- the font color of input fields (thanks to the **fieldsFontColor** attribute).

```
vads_iframe_options = {"fieldsBackgroundColor":"#000000","fieldsFontColor":"#FFFFFF"}
```

Figure 5: Default input fields

Figure 6: Custom input fields

8. TRANSMITTING THE RETURN URLS

When iframe mode is enabled, there is no button displayed on the payment page to abandon the payment.

The merchant web site has to manage:

- redirection in case of successful payment,
- redirection in case of declined payment,
- redirection in case the payment is abandoned by the buyer

In the first two cases, to automatically redirect the buyer to the merchant website, populate the following fields:

- **vads_redirect_success_timeout** and **vads_redirect_error_timeout** to "0",
- **vads_url_success** with the **URL of your order confirmation page**,
- **vads_url_return** with the URL of your return page in case of rejected payment,
- **vads_return_mode** with the **POST** or **GET** value to retrieve the payment result and display it on your order confirmation page if necessary.



The **GET** value does not allow to specify the display mode of the return page.

To manage the case of abandoned payment, you can add a button on top of the iframe for closing it.

```
...
<div style="float:right;">
  <input type="image" src="close.png" height="30" width="30"
  onclick="window.parent.removeIframe();" />
</div>
<iframe name="nameFrame" src="payment_form.php"
...

```

If you want to be notified when payments are canceled, make sure you have configured and enabled the notification rule “**Instant Payment Notification URL on cancellation**” in your Merchant Back Office.

9. TRANSMITTING THE TRANSACTION RECEIPT TO THE BUYER

When iframe mode is enabled, the buyer cannot view the transaction receipt on the summary page.

The receipt will automatically be sent by mail (provided that the notification rule “*Payment confirmation e-mail sent to the buyer*” is enabled in your Merchant Back Office).

The buyer's e-mail is required when using iframe mode.

Use the **vads_cust_email** field in your payment form to transmit the buyer's email address.

To use this service, contact the E-commerce customer service to enable the necessary functions within your offer.

10. SPECIFYING THE DISPLAY MODE OF THE RETURN PAGE

The **FORM_TARGET** attribute of the **vads_theme_config** field allows to define how the return page will be displayed.



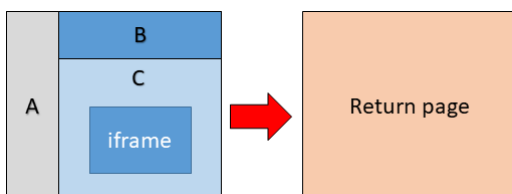
You must make sure that you set the **vads_return_mode** field to **POST** so that the display mode of the return page is taken into account.

10.1. Displaying the return page in full screen

Use the **_top** value of the **FORM_TARGET** attribute.

Syntax: **vads_theme_config="FORM_TARGET=_top**

Result:

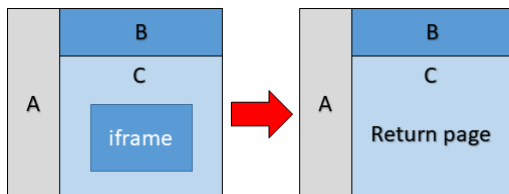


10.2. Displaying the return page in the parent frame

Use the **_parent** value of the **FORM_TARGET** attribute.

Syntax: `vads_theme_config="FORM_TARGET=_parent`

Result:

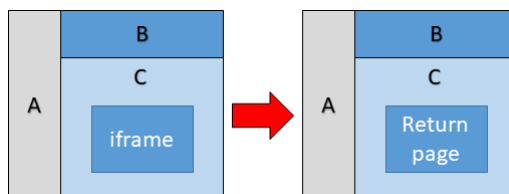


10.3. Displaying the return page in the same iframe

Use the **_self** value of the **FORM_TARGET** attribute.

Syntax: `vads_theme_config="FORM_TARGET=_self`

Result:



11. CATCHING THE END OF PAYMENT

For integrations including return within the same iframe (**FORM_TARGET=_self**), it is required to close the iframe and display the payment result in the parent frame.

To do this, you can declare the following JS function in the parent frame:

```
window.removeIframe = function() {  
  var iframe = document.querySelector('iframe[name=payment]');  
  if (iframe) {  
    iframe.remove();  
  }  
};
```

Insert the following code in your return pages to call the function:

```
<script>  
  parent.removeIframe();  
</script>
```

12. OFFERING ADDITIONAL PAYMENT ATTEMPTS

With full-screen redirection, the payment gateway can offer the "other payment attempt" button if the payment is declined (if the number of attempts is set to zero in the configuration).

The number of attempts is specified in your Merchant Back Office.

When iframe mode is enabled, this behavior is not available.

You can nevertheless propose a new attempt on your return page (URL defined in the **vads_url_return** field) and display the iframe containing the payment page again.

Sample code on the return page:

```
...  
<div>  
  <p> Payment declined . Would you like to try again?</p>  
  
  <input type="button" value="New attempt" onclick="window.parent.refreshIframe();">  
  
  <input type="button" value="Cancel" onclick="window.parent.removeIframe();">  
</div>  
...
```

Source code of the refreshIframe() function to be added on the page containing the iframe:

```
function refreshIframe(){  
  var iframe = document.getElementById('nameFrame');  
  iframe.src = iframe.src;  
}
```

13. MANAGING TIMEOUTS

A payment session is the time given to the buyer for making a payment.

The payment session delay is 10 minutes.

Once the session has expired, the buyer is disconnected. The merchant web site is notified if the “**Instant Payment Notification URL on cancellation**” notification rule is configured and enabled.

If the buyer makes a transaction on the payment page after the expiry of the payment session, an error message will be displayed.

To avoid this logout message, you can add a counter to your page.

After some time (5 minutes for example) the iframe will be closed automatically.

```
...
    setTimeout(window.parent.removeIframe(), '30000');
</body>
</html>
```

Another solution consists in capturing the “message” event sent by the payment gateway in case of disconnection.

For this, paste the code below into the page that invokes the iframe:

```
if (window.addEventListener) {
    window.addEventListener("message", onMessage, false);
} else if (window.attachEvent) {
    window.attachEvent("onmessage", onMessage, false);
}
function onMessage(event) {
    if (event.origin !== "https://paiement.systempay.fr") return;
    var data = event.data;
    if (typeof(window[data.func]) == "function") {
        window[data.func].call();
    }
}
```

Then, implement the **handlePaymentDisconnection()** function:

```
function handlePaymentDisconnection() {
    //Close iframe
    //Show message
    //Show button to relaunch payment
    //etc, etc
}
```

Make sure you respect the spelling of the function name.