

Iframe integration



Advantages

This option allows the buyer to stay on a known and trusted environment.

The payment is fluid and cart abandonment are rare.



It is possible to use a custom template if payment page integration is used.

For more details: [Implementation of a Custom Template](#)



The iframe is web responsive and can be integrated in a mobile application.

Table of Contents

- [iFrame implementation summary](#)
- [Setting up the iframe and configuring the URL passed to the src attribute](#)
 - [iframe & customField14 tag](#)
 - [Settings: Height & Width](#)
- [Creation of the 'Success' and 'Failure' pages](#)
- [Implementation of redirection in JavaScript](#)
 - [Intégration of postMessage method in redirect-success and redirect-failure](#)
 - [Integrating the Event Listener into the payment page](#)
- [Dynamic customization](#)

iFrame implementation summary

- Setting up the iframe in the *merchant's checkout page*
- Configuring the URL passed to the src attribute
 - Preparing URLSuccess and URLFailure parameters with successful and error return URLs
 - Placing these 2 parameters in the hash of the Data parameter
 - Passing Data as a parameter of the URL provided to the src attribute of the iframe
 - Added the CustomField14=iframe parameter to have only the display of the payment block
- Creation of the 'Success' and 'Failure' pages
 - Preparing the HTML code for redirect-success and redirect-failure
- Implementation of redirection in JavaScript
 - Integration of the postMessage method in redirect-success and redirect-failure
 - Integrating the Event Listener into the *merchant's checkout page*

Setting up the iframe and configuring the URL passed to the src attribute

iframe & customField14 tag

An iframe is an HTML tag that allows you to embed one HTML page into another.

To do this, you must use the **<iframe> tag** and **specify the URL** of the page you want to display in the **src attribute**. The checkout page URL must contain the following value: **customField14=iframe**.

Exemple d'intégration du formulaire carte au sein d'une page HTML

```
<!DOCTYPE html>
<HTML lang="en">
  <HEAD>
    <META charset="UTF-8" />
    <META http-equiv="X-UA-Compatible" content="IE=edge" />
    <META name="viewport" content="width=device-width, initial-scale=1.0" />
    <TITLE>Site marchand - commande</TITLE>
  </HEAD>
  <BODY>
    ...
    <IFRAME
      src="https://paymentpage.axepta.bnpparibas/payssl.aspx?
MerchantID=YourMID&Len=123&Data=AGSDJ...ASDF&CustomField14=iframe"
    />
    ...
  </BODY>
</HTML>
```



Warning

It will be necessary to pass the parameters required to the payment with this URL

At a minimum: MerchantID, Data, and Len

Settings: Height & Width

In pixel

```
<IFRAME
  src="https://paymentpage.axepta.bnpparibas/payssl.aspx?
MerchantID=YourMID&Len=123&Data=AGSDJ...ASDF&CustomField14=iframe"
  height="600"
  width="800"
/>
```

In CSS

```
#payment-iframe {
  height: 600px;
  width: 800px;
}

// ou pour un rendu fullscreen
#payment-iframe {
  height: 100vh;
  width: 100vw;
}

<IFRAME
  id="payment-iframe"
  src="https://paymentpage.axepta.bnpparibas/payssl.aspx?
MerchantID=YourMID&Len=123&Data=AGSDJ...ASDF&CustomField14=iframe"
/>
```

Creation of the 'Success' and 'Failure' pages

When the bearer submits the payment form, a request is sent to the Axepta server.

If the payment is validated, a successful response is sent. Otherwise, an Error response will be issued.

The goal is to display a success or failure page instead of the iFrame based on the payment result.



Note: The server returns an HTTP response with the POST method. We must therefore create a route on the server of the merchant site to process this answer.

If you want to redirect the user to a page of the merchant site displayed instead of the iframe, it will be necessary to provide a redirect page.

- URLSuccess: <https://mon-site.com/redirect-success>
- URLFailure: <https://mon-site.com/redirect-failure>

The merchant creates these pages that will be displayed when the bearer is redirected.

Each will run a script that will redirect the user to the page in success or payment error.

The implementation of this script is indicated in the next section.

Implementation of redirection in JavaScript

Intégration of postMessage method in redirect-success and redirect-failure

The `postMessage` solution "enables secure cross-domain communication".

The goal is to "communicate" with the parent page. This communication will be done through a MessageEvent type event

Code de la page de redirection redirect-success

```
<!DOCTYPE html>
<HTML lang="en">
  <HEAD>
    <META charset="UTF-8" />
    <META http-equiv="X-UA-Compatible" content="IE=edge" />
    <META name="viewport" content="width=device-width, initial-scale=1.0" />
    <TITLE>Site marchand - redirection</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT>
      function sendMessage() {
        // we target the parent page with window.parent
        // and send it a postMessage window.parent.postMessage(
        // message in character string
        // "payment success",
        // we specify the message origin
        "https://my-website.com"
      };
    }

    sendMessage();
  </SCRIPT>
</BODY>
</HTML>
```



The failure page must also be created.

Integrating the Event Listener into the payment page

In the page that contains the iframe, we will set up an event listener. It will be responsible for performing a function when it receives a message event.

Modification du code de la page qui int gre l'iframe - Ajout EventListener

```
<!DOCTYPE html>
<HTML lang="en">
  <HEAD>
    <META charset="UTF-8" />
    <META http-equiv="X-UA-Compatible" content="IE=edge" />
    <META name="viewport" content="width=device-width, initial-scale=1.0" />
    <TITLE>Site marchand - commande</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT>
      // sur la page courante
      // on place un  couteur avec la m thode addEventListener
      window.addEventListener(
        // type d' v nement    couter
        "message",
        // fonction de retour (callback) qui sera ex cut e
        // quand un  v nement de type "message" sera  mis
        (event) => {
          // Si l'origin n'est pas la m me que celle plac e dans le message
          // on arr te l'ex cution de la fonction
          if (event.origin !== "https://mon-site.com") return;

          // on redirige l'utilisateur sur la page souhait e
          window.location.href = "https://mon-site.com/payment-success";
        }
      );
    </SCRIPT>
    ...
    <IFRAME
      src="https://paymentpage.axepta.bnpparibas/payssl.aspx?
MerchantID=YourMID&Len=123&Data=AGSDJ...ASDF&CustomField14=iframe"
    />
    ...
  </BODY>
</HTML>
```

Dynamic customization

Other custom fields can be added to the request in order to dynamically change the look and feel of the iframe.

Please find below the fields available :

- CustomField100 : Background color
- CustomField101 : Colour of button text when button is enabled
- CustomField102 : Colour of button background when button is enabled
- CustomField103 : Color of button text on hover when button is enabled

- CustomField104 : Colour of button background on hover when button is enabled
- CustomField105 : Colour of button text when button is disabled
- CustomField106 : Colour of button background when button is disabled



The **hexadecimal** colour values must be used in the CustomField fields and the **'#' symbol must be encoded** so it will be replaced by the value `'%23'`.

For example, to display a red button (when the button is enabled), the Customer Field102 must be enhanced as follows: CustomField102=%23FF0000.