iFrames-integratie

Voordelen van iFrame

- **Vertrouwde omgeving**: De koper blijft op de **website van de handelaar**, in een bekende en veilige omgeving.
- **Vloeiend betaalproces**: De betalingsstroom verloopt **naadloos**, zonder omleiding naar een externe pagina.

Iframe-integratie behoudt PCI-compliancy, omdat de betalingsgegevens veilig via Axepta BNP Paribas worden verwerkt.
Zorg voor een responsief ontwerp om de iframe correct op mobiele apparaten weer te geven

Inhoudstabel

- Samenvatting van de iframe-implementatie
  - Implementatie van de iframe in de betaalpagina van de handelaar
- Implementatie van de iframe en configuratie van de URL in het src-attribuut
  - Iframe tag & customField14
  - Instellingen: Hoogte & Breedte
- Aanmaken van 'Return'-pagina's
- Omleiding instellen met JavaScript
  - Integratie van postMessage in redirect-success en redirect-failure
  - Integratie van de Event Listener in de betaalpagina
- Dynamische aanpassingen met extra CustomFields
  - Samenvatting

Samenvatting van de iframe-implementatie

Implementatie van de iframe in de betaalpagina van de handelaar

- **Configuratie van URL's in het `"urls"`**-object van de betaalverzoek-API**
- **Voorbereiding van de `urls.return` en `urls.cancel`-parameters**
  - Opgeven van URL's voor **geslaagde** en **afgebroken betalingen**.
- **Plaatsing van deze parameters in de `Data`-parameter-hash**
- **Doorgeven van `Data` als parameter in de iframe-`src`-URL**
- **Toevoegen van `CustomField14=iframe`**
  - Zorgt ervoor dat **alleen het betaalblok** wordt weergegeven.
- **Aanmaken van 'Succes'- en 'Fout'-pagina's**
- **Voorbereiding van de HTML-code voor:**
  - `redirect-success` (bij geslaagde betaling)
  - `redirect-failure` (bij mislukte betaling)
- **Implementatie van JavaScript-omleidingen**
- **Integratie van de `postMessage`-methode**
  - Voor communicatie tussen iframe en handelaarsite bij `redirect-success` en `redirect-failure`.
- **Toevoegen van een *Event Listener* in de betaalpagina van de handelaar**
  - Om berichten van de iframe op te vangen (bv. betalingsstatus).

Implementatie van de iframe en configuratie van de URL in het src-attribuut

Iframe tag & customField14

Een **iframe** (Inline Frame) is een **HTML-tag** die het mogelijk maakt om **een externe webpagina in te bedden** binnen een bestaande pagina. Dit wordt vaak gebruikt om content van derden (zoals betaalpagina's) naadloos te integreren zonder de gebruiker naar een andere site te sturen.

Om de **betaalpagina van Axepta BNP Paribas** in te bedden, gebruik je de `<iframe>`-tag met de volgende vereisten:

De **`src`-URL** van de iframe **moet** de volgende parameter bevatten:

`customField14=iframe`

Voorbeeld: Integratie van het kaartbetalingsformulier in een HTML-pagina

```
<!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>Merchant website - Order</TITLE>
  </HEAD>
  <BODY>
    ...
    <IFRAME
      src="https://paymentpage.axepta.bnpparibas/payssl.aspx?token=YOUR_PAYMENT_TOKEN"
    />
    ...
  </BODY>
</HTML>
```

Instellingen: Hoogte & Breedte

In pixels

```
<IFRAME
  src="https://paymentpage.axepta.bnpparibas/payssl.aspx?token=YOUR_PAYMENT_TOKEN"
  height="600"
  width="800"
/>
```

In CSS

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

// or for a fullscreen display
#payment-iframe {
  height: 100vh;
  width: 100vw;
}
<IFRAME
  id="payment-iframe"
  src="https://paymentpage.axepta.bnpparibas/payssl.aspx?token=YOUR_PAYMENT_TOKEN"
/>
```

Aanmaken van 'Return'-pagina's

Wanneer de koper het betaalformulier indient, wordt een verzoek naar de **Axepta BNP Paribas-server** gestuurd. Afhankelijk van de actie van de gebruiker wordt het antwoord verzonden naar:

Scenario	Doel-URL	Beschrijving
Betaling voltooid (succes/mislukt)	`urls.return`	Ontvangt een callback met de betalingsstatus. Toont een **bevestigingspagina**.
Betaling geannuleerd	`urls.cancel`	Wordt getriggerd als de koper het proces **afbreekt**. Toont een **annuleringspagina**.

Doel: Vervang de iframe door een **succes- of foutpagina**, afhankelijk van het betalingsresultaat.

Belangrijke opmerking:

Een `return`-callback garandeert geen geslaagde betaling! Je **moet altijd** de transactiedetails opvragen via de **`getByPayId` API** om de **daadwer kelijke status** te verifiëren.

Voor gedetailleerde informatie over:

- Werking van `urls.return` en `urls.cancel`
- Verwerking van betalingsnotificaties
- Webhooks en server-side callbacks

Zie documentatie: Callback and Payment Notification Management

> Note: De server van Axepta BNP Paribas retourneert een HTTP-antwoord met de POST-methode. Daarom moet de handelaar een server-route aanmaken om dit antwoord te verwerken.

Als je de gebruiker wilt **doorsturen naar een pagina op de handelaarsite** (in plaats van de iframe te tonen), moet je **twee omleidingspagina's** voorb ereiden:

Axepta URL	`urls.return`	`urls.cancel`
`getByPayId`-status	OK (geslaagd)	ERROR (mislukt)
Omleiding	**Succesvolle betaling**	**Mislukte betaling**

- De handelaar moet **twee pagina's** maken die worden getoond na omleiding:
  1. **Succesvolle betaling**
     - **URL-voorbeeld:** `https://uw-website.be/betalings-succes`
     - Toont een bevestiging (bv. *"Betaling voltooid!"*).
  2. **Mislukte/geannuleerde betaling**
     - **URL-voorbeeld:** `https://uw-website.be/betalings-fout`
     - Toont een foutmelding (bv. *"Betaling mislukt, probeer opnieuw."*).

Elke pagina (**succes/fout**) moet een **script uitvoeren** dat de gebruiker doorstuurt naar de definitieve bestemmingspagina. De **configuratie van dit script** wordt beschreven in de volgende sectie.

Omleiding instellen met JavaScript

Integratie van postMessage in redirect-success en redirect-failure

Om de gebruiker na een betaling **dynamisch om te leiden** naar de juiste pagina (succes/fout), maken we gebruik van de **`postMessage`-methode**. Deze methode maakt **veilige communicatie tussen domeinen** mogelijk, zodat de iframe met de hoofdpagina kan "praten" via een **`MessageEvent`**.

Handle returnUrl (success - failed) backend integration

```
const express = require('express');

const app = express();
const PORT = 3000;

const API_BASE_URL = 'https://paymentpage.axepta.bnpparibas';
const BEARER_TOKEN = 'YOUR_BEARER_TOKEN_HERE';

// Template function to generate HTML response based on payment status
function renderPaymentPage(payId, status, isSuccess) {
  const title = isSuccess ? 'Payment Successful' : 'Payment Failed';
  const message = isSuccess ? 'Payment Successful!' : 'Payment Failed';
  const eventType = isSuccess ? 'PAYMENT_SUCCESS' : 'PAYMENT_FAILED';
  const redirectUrl = isSuccess
    ? `/order/confirmation?payId=${payId}`
    : `/payment/retry?payId=${payId}`;

  return `
    <!DOCTYPE html>
    <html>
    <head>
      <title>${title}</title>
    </head>
    <body>
      <h1>${message}</h1>
      <p>Transaction ID: ${payId}</p>
      <script>
        // If page is loaded inside an iframe, notify the parent window
        if (window.parent !== window) {
          window.parent.postMessage({
            type: '${eventType}',
            payId: '${payId}',
            status: '${status}'
          }, '*');
        }

        // Redirect user after 3 seconds
        setTimeout(() => {
          window.location.href = '${redirectUrl}';
        }, 3000);
      </script>
    </body>
    </html>
  `;
}

// Return URL endpoint - This is where Axepta redirects customers after payment
app.get('/payment/return', async (req, res) => {
  try {
    // Step 1: Extract PayID from URL query parameters
    // Example: /payment/return?PayID=abc123
    const payId = req.query.PayID;

    // Validate that PayID exists
    if (!payId) {
      return res.status(400).send('PayID is missing');
    }

    // Step 2: Call Axepta API to retrieve payment details
    const response = await fetch(
      `${API_BASE_URL}/api/v2/payments/getByPayId/${payId}`,
      {
        method: 'GET',
        headers: {
```

```
          'Authorization': `Bearer ${BEARER_TOKEN}`,
          'Content-Type': 'application/json'
        }
      }
    );

    // Check if API request was successful
    if (!response.ok) {
      throw new Error(`API returned status ${response.status}`);
    }

    // Parse JSON response from API
    const paymentData = await response.json();
    const status = paymentData.status;

    // Step 3: Determine if payment was successful
    const isSuccess = status === 'SUCCESS' || status === 'AUTHORIZED';

    // Step 4: Send HTML response to customer
    res.send(renderPaymentPage(payId, status, isSuccess));

  } catch (error) {
    // Log error for debugging
    console.error('Error retrieving payment details:', error.message);

    // Return simple error message to customer
    res.status(500).send('Server error');
  }
});

// Start the server
app.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});
```

Integratie van de Event Listener in de betaalpagina

Op de **pagina die de iframe bevat**, moet een **`Event Listener`** worden toegevoegd om berichten van de iframe op te vangen en actie te ondernemen (bv. omleiden naar een bevestigingspagina).

Aanpassing van de iframe-integratiecode - Addition of 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>Merchant website - order</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT>
      // on the current page
      // we place a listener with the addEventListener method
      window.addEventListener(
        // event type to listen to
        "message",
        // callback function that will be executed
        // when an event of type "message" will be emitted
        (event) => {
          // If the origin is not the same as the one placed in the message
          // we stop the execution of the function
          if (event.origin !== "https://mon-site.com") return;

          // we redirect the user to the desired page
                          }
                        );
          window.location.href = "https://mon-site.com/payment-success";
        }
      );
    </SCRIPT>
    ...
    <IFRAME
      src="IFRAME src="https://paymentpage.axepta.bnpparibas/payssl.aspx?token=YOUR_PAYMENT_TOKEN"
    />
    ...
  </BODY>
</HTML>
```

Dynamische aanpassingen met extra CustomFields

Je kunt de **stijl van de iframe** dynamisch wijzigen met extra `CustomFields`. Hieronder vind je de beschikbare velden voor **kleurinstellingen** (gebruik **hexadecimale kleurcodes** met `%23` in plaats van #):

Veld	Beschrijving	Voorbeeld
`CustomField100`	Achtergrondkleur van de betaalpagina.	`CustomField100=%23FFFFFF` (wit)
`CustomField101`	Tekstkleur van **actieve knoppen**.	`CustomField101=%23000000` (zwart)
`CustomField102`	Achtergrondkleur van **actieve knoppen**.	`CustomField102=%23FF0000` (rood)
`CustomField103`	Tekstkleur bij **hover** op actieve knoppen.	`CustomField103=%23FFFFFF` (wit)
`CustomField104`	Achtergrondkleur bij **hover** op actieve knoppen.	`CustomField104=%23CC0000`
`CustomField105`	Tekstkleur van **inactieve knoppen**.	`CustomField105=%23999999` (grijs)
`CustomField106`	Achtergrondkleur van **inactieve knoppen**.	`CustomField106=%23E0E0E0`

Samenvatting

Stap	Actie

1. `postMessage` in omleidingspagina's	Stuur betalingsstatus naar de hoofdpagina.
2. `Event Listener` in hoofdpagina	Ontvang berichten en leid door naar succes/foutpagina.
3. Stijl aanpassen	Gebruik `CustomField100-106` voor kleuren.
4. Testen	Controleer de communicatie tussen iframe en hoofdpagina in de sandbox.

- **Veiligheid:** Controleer altijd `event.origin` om **cross-site scripting (XSS)** te voorkomen.
- **Kleurcodes:** Gebruik `%23` in plaats van # (bv. `%23FF0000` voor rood).