Set the language when generating the card component

Overview

Set up the 2Pay.js drop-in payments client to accept payments on your website. To collect payments you need to create a payment form, tokenize sensitive card information, and use that token to create a charge.

Use the below method to set the language when generating the card component.

Use Case

1. Create a form with id="payment-form":

   ```html
   <form type="post" id="payment-form">
   <div class="form-group">
     <label for="name" class="label control-label">Name</label>
     <input type="text" id="name" class="field form-control">
   </div>
   <button class="btn btn-primary" type="submit">Generate token</button>
   </form>
   ```

2. Inside the form, add an element with id="card-element". The form should now look like this:

   ```html
   <form type="post" id="payment-form">
   <div class="form-group">
     <label for="name" class="label control-label">Name</label>
     <input type="text" id="name" class="field form-control">
   </div>
   <div id="card-element">
     <!-- A TCO IFRAME will be inserted here. -->
   </div>
   <button class="btn btn-primary" type="submit">Generate token</button>
   </form>
   ```

3. Include the 2Pay.js drop-in payments client JavaScript:

   ```html
   <script type="text/javascript" src="https://2pay-js.2checkout.com/v1/2pay.js"></script>
   ```

4. Insert the following configuration JavaScript to set the language when adding the card component, and generate the
token request:

```javascript
window.addEventListener('load', function() {
  // Initialize the JS Payments SDK client.
  let jsPaymentClient = new TwoPayClient('AVLRNG');

  // Set the desired language to be used inside the iframe.
  jsPaymentClient.setup.setLanguage('ro');

  // Create the component that will hold the card fields.
  let component = jsPaymentClient.components.create('card');

  // Mount the card fields component in the desired HTML tag. This is where the iframe will be located.
  component.mount('#card-element');

  // Handle form submission.
  document.getElementById('payment-form').addEventListener('submit', (event) => {
    event.preventDefault();

    // Extract the Name field value
    const billingDetails = {
      name: document.querySelector('#name').value
    };

    // Call the generate method using the component as the first parameter
    // and the billing details as the second one
    jsPaymentClient.tokens.generate(component, billingDetails).then((response) => {
      console.log(response.token);
    }).catch((error) => {
      console.error(error);
    });
  });
});
```

5. Place the order using the generated token.

**Demo**

After performing the steps above, your implementation should look like this:

```
Media, iframe, embed and object tags are not supported inside of a PDF.
```