PepperPay Shopping Cart

Basic Usage

Step 1: Include the Checkout Script

Embed the PepperPay checkout script within the HEAD tag of your webpage. This script is required to initialize and configure the PepperPay checkout system in your application.

```
<script src="https://portal.pepperpay.com/checkout/checkout.js"></script>
```

Step 2: Initialize the Checkout Button

Add the following script before or after closing the body tag to your page which initializes this button.

```
<script>
const buttonInjector = new ButtonInjector("containerId", "priceTagId", "apiKey", "returnUrl",
"cssUrl");
</script>
```

The parameters within the ButtonInjector function in the script are explained below:

- **containerId:** The ID of the HTML element where the checkout button will be inserted.
- **priceTagId:** The ID of the HTML element containing the user's total purchase price. *Only currency format values (with decimals) are allowed, such as "9.99", "99.99", "999.99".*
- **apiKey:** This is your unique API key provided by PepperPay, which authenticates your application with the PepperPay systems. Get your API key here.
- **returnUrl:** This is the return URL after the checkout is processed. Result can be *"success"*, *"cancel"* or *"failure"*. More information below in Response Handling section.
- **cssUrl (optional):** A public URL for a CSS stylesheet that customizes the appearance of the payment page.

The checkout button when clicked redirects users to the checkout page.

Congratulations! You've successfully implemented the PepperPay checkout button.

Now, your users will be redirected to the PepperPay checkout page upon clicking this button.

Response Handling

Return Response

Once the payment is done successfully, the checkout page will redirect to the returnUrl passed in the Button Injector.

- **On Success:** `returnUrl + ?result=success` (e.g. www.your-website.com?result=success)

  This includes declined transactions
- **On Cancel:** `returnUrl + ?result=cancel` (e.g.www.your-website.com?result=cancel)
- **On Failure:** `returnUrl + ?result=failure&message=error message` (e.g. www.your-website.com?result=failure&message=errormessage)

WooCommerce Integration (WordPress)

This guide will provide step by step instructions on how to integrate a payment button into your WooCommerce store.

Step 1: Navigate to Your Project Folder

Go to your project folder on your hosting server. Open the `functions.php` file, which you can find under the path: `wp-content/themes/your-theme/`.

Step 2: Inject the Checkout Script

In the `functions.php` file, include the checkout script by adding the following function:

```
function  button_injector_script(){
echo
'<script src="https://portal-dev.pepperpay.com/checkout/test-checkout.js"></script>';
}
add_action('wp_head','button_injector_script');
```

Step 3: Initialize the Checkout Button

This step is critical. You must add the checkout button initialization to the footer `wp_footer`.

Include the following function in the `functions.php` file:

```
function  button_instance_script(){
echo
'<script>const buttonInjector = new ButtonInjector("containerId", "priceTagId", "apiKey",
"returnUrl", "cssUrl")</script>';
}
add_action('wp_footer','button_instance_script');
```

For more information about **containerId**, **priceTagId**, **apiKey**, **returnUrl** and **cssUrl**, read Basic

Usage.

Step 4: Implement the Button

From your website dashboard, edit the page where you want to insert the button. Use the

following HTML code block:

```
<div id="containerId"></div>
<p id="priceTagId">9.99</p>
```

The button should now be integrated into your WooCommerce store.

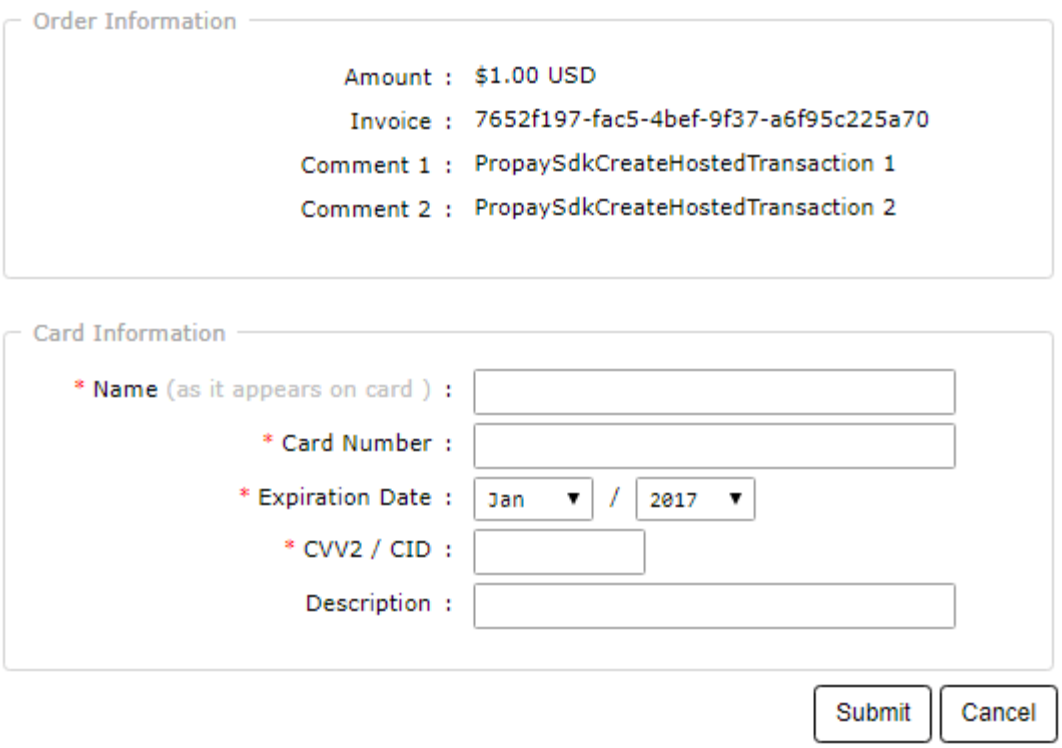
Customizing the appearance of the payment page

Default CSS File

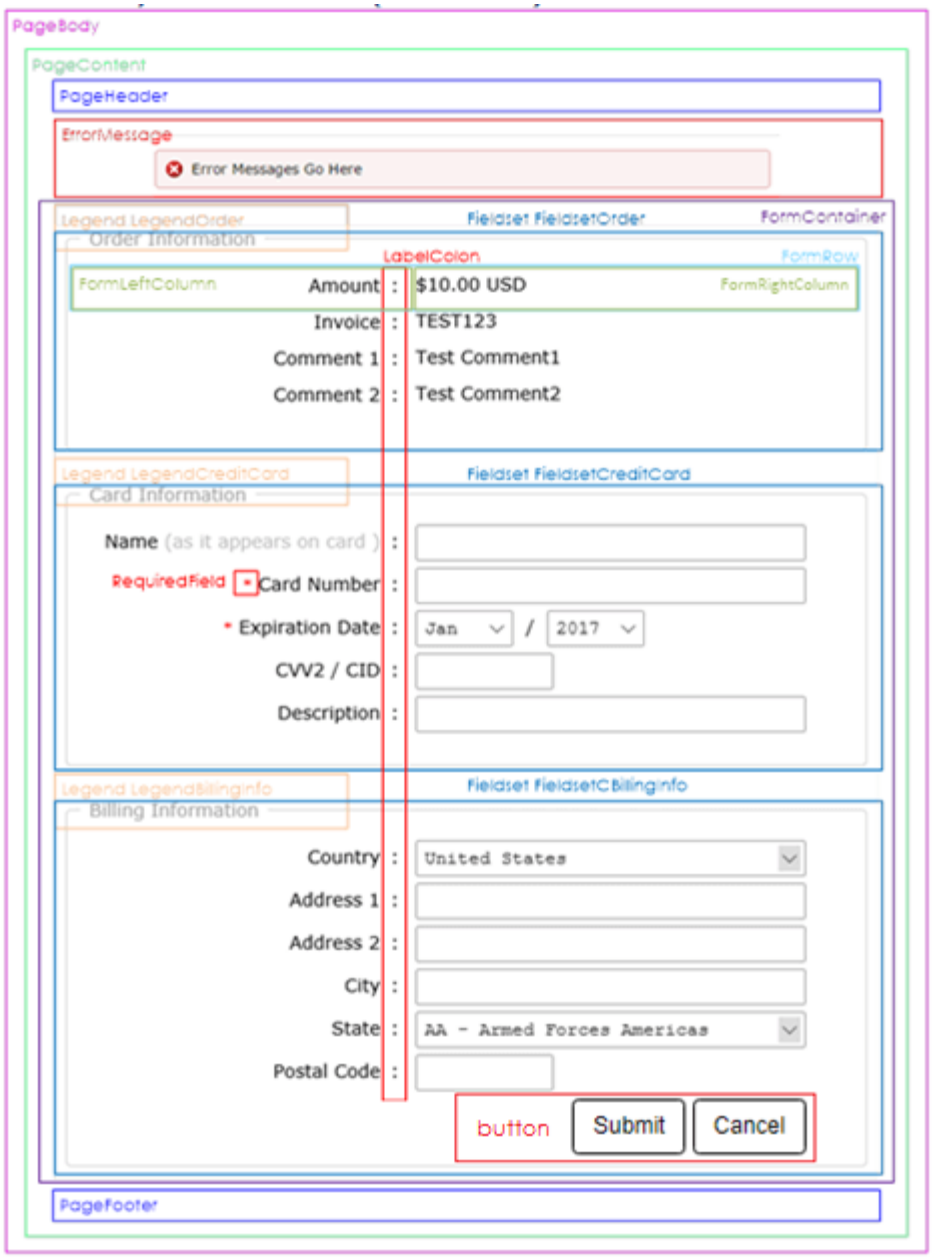
This is the default CSS file which you can use to help build your own:

https://portal.pepperpay.com/checkout/checkout.css

Default Appearance of Credit Card Transactions



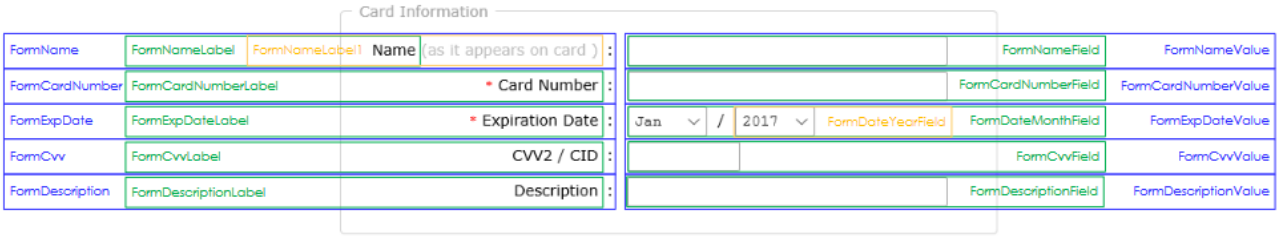
CSS Style Section Guide - Credit Card



CSS Style Element Guide - Order Information

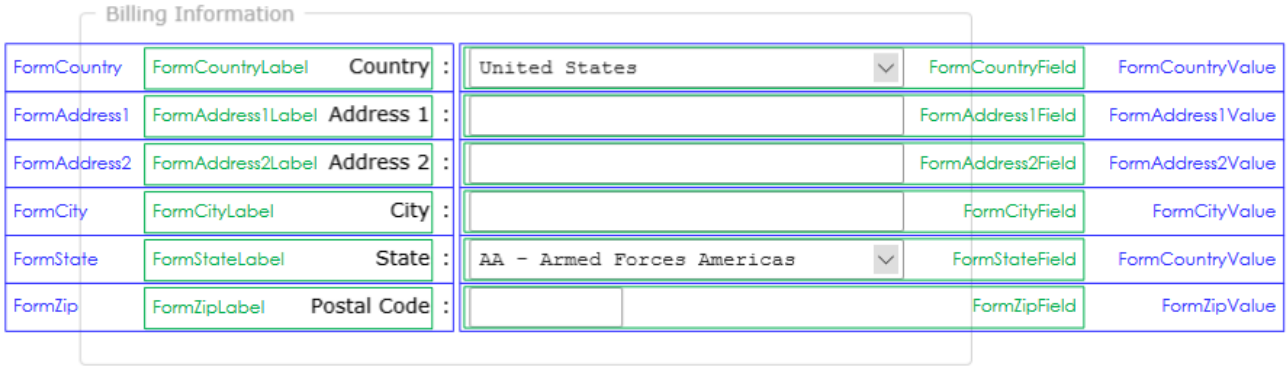
- All Labels inherit the "Label" class

CSS Style Element Guide - Credit Card Information



- All Labels inherit the "Label" class
- All TextFields inherit the "TextBox" Class
- All Drop Down Lists inherit the "DropDownClass"

CSS Style Element Guide - Billing Information



- All Labels inherit the "Label" class
- All TextFields inherit the "TextBox" Class
- All Drop Down Lists inherit the "DropDownClass"

CSS Style Element Guide - Buttons

- All Buttons inherit the "button" class