To connect a Stripe card reader to your web application using **Core PHP** and **jQuery**, you need to use the Stripe Terminal API and the Stripe Terminal JavaScript SDK. Here's a step-by-step guide:

**Step 1: Set Up Your Stripe Account**

1. Log in to your [Stripe Dashboard](https://dashboard.stripe.com/).
2. Enable **Terminal** in your account settings.
3. Register a **reader** in the Terminal section of the dashboard.
   * You'll need to pair your reader via the dashboard or during runtime.

**Step 2: Install Stripe PHP SDK**

Install the Stripe PHP library via Composer:

bash

CopyEdit

composer require stripe/stripe-php

If you're not using Composer, download the Stripe PHP SDK manually and include it in your project.

**Step 3: Backend Endpoint for Stripe Connection**

Create an endpoint in your backend (PHP) to handle client secrets for the reader connection and payments.

**Example: stripe\_backend.php**

php

CopyEdit

<?php

require 'vendor/autoload.php'; // Include the Stripe PHP SDK

\Stripe\Stripe::setApiKey('sk\_test\_your\_secret\_key'); // Replace with your Stripe Secret Key

// Handle the frontend request

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['action'])) {

$action = $\_POST['action'];

if ($action === 'create\_connection\_token') {

// Create a connection token for the card reader

$connectionToken = \Stripe\Terminal\ConnectionToken::create();

echo json\_encode(['secret' => $connectionToken->secret]);

exit;

}

if ($action === 'create\_payment\_intent') {

$amount = intval($\_POST['amount']);

$currency = $\_POST['currency'];

// Create a PaymentIntent

$paymentIntent = \Stripe\PaymentIntent::create([

'amount' => $amount,

'currency' => $currency,

'payment\_method\_types' => ['card\_present'],

'capture\_method' => 'manual',

]);

echo json\_encode(['client\_secret' => $paymentIntent->client\_secret]);

exit;

}

}

http\_response\_code(400);

echo json\_encode(['error' => 'Invalid request']);

?>

**Step 4: Frontend Integration**

Use **jQuery** and **Stripe Terminal JavaScript SDK** for client-side operations.

**Example: index.html**

html

CopyEdit

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Stripe Terminal Integration</title>

<script src="https://js.stripe.com/v3/"></script>

<script src="https://unpkg.com/@stripe/terminal-js"></script>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

</head>

<body>

<button id="connect-reader">Connect Reader</button>

<button id="process-payment">Process Payment</button>

<script>

const terminal = StripeTerminal.create({

onFetchConnectionToken: async () => {

// Fetch the connection token from your backend

const response = await $.post('stripe\_backend.php', { action: 'create\_connection\_token' });

const data = JSON.parse(response);

return data.secret;

},

});

let reader;

// Step 1: Connect to the card reader

$('#connect-reader').click(async () => {

const discoverResult = await terminal.discoverReaders();

if (discoverResult.error || discoverResult.discoveredReaders.length === 0) {

alert('No readers found');

return;

}

reader = discoverResult.discoveredReaders[0]; // Use the first reader found

const connectResult = await terminal.connectReader(reader);

if (connectResult.error) {

alert(`Failed to connect: ${connectResult.error.message}`);

} else {

alert('Reader connected');

}

});

// Step 2: Process Payment

$('#process-payment').click(async () => {

const amount = 5000; // Amount in cents (e.g., $50.00)

const currency = 'usd';

// Create a payment intent on the backend

const response = await $.post('stripe\_backend.php', {

action: 'create\_payment\_intent',

amount: amount,

currency: currency,

});

const { client\_secret } = JSON.parse(response);

const paymentResult = await terminal.collectPaymentMethod(client\_secret);

if (paymentResult.error) {

alert(`Payment failed: ${paymentResult.error.message}`);

} else {

const confirmResult = await terminal.processPayment(paymentResult.paymentIntent);

if (confirmResult.error) {

alert(`Payment confirmation failed: ${confirmResult.error.message}`);

} else {

alert('Payment successful!');

}

}

});

</script>

</body>

</html>

**Step 5: Test Your Integration**

1. Connect your Stripe card reader to your application.
2. Use the "Connect Reader" button to establish a connection.
3. Use the "Process Payment" button to collect payment via the reader.

**Notes**

* **Reader Compatibility**: Ensure you're using a supported Stripe card reader like **BBPOS WisePOS E**.
* **Environment**: Use sk\_test\_ keys for testing and switch to sk\_live\_ keys for production.
* **Security**: Never expose secret keys on the frontend. Always fetch sensitive data via your backend.