```
<!--Braintree integration-->
{braintree_css}

{literal}
<!-- Load the client component. -->
<script src="https://js.braintreegateway.com/web/3.100.0/js/
client.min.js"></script>
<script src="https://www.paypalobjects.com/api/checkout.js" data-
version-4 log-level="warn"></script>
<script src="https://js.braintreegateway.com/web/3.100.0/js/data-
collector.min.js"></script>
<script src="https://js.braintreegateway.com/web/3.100.0/js/paypal-
checkout.min.js"></script>
<!-- Load the 3D Secure component. -->
<script src="https://js.braintreegateway.com/web/3.100.0/js/three-d-
secure.min.js"></script>
<!-- Load the Hosted Fields component. -->
<script src="https://js.braintreegateway.com/web/3.100.0/js/hosted-
fields.min.js"></script>




<script language="JavaScript">
$(document).ready(function(){

    if ($('#tokenization-key').length) {
        coresense_initialize_payment_processor();
    }


    $('#payment_types_select').on('change', function() {
        var selected_str = $(this).children("option:selected").text();
        if(selected_str == 'Paypal') {
            $(".paginated-checkout-controls > a.right").css("display",
"none");
            $('.paginated-checkout-navigation > :last-child').hide();
        } else {
            $(".paginated-checkout-controls > a.right").css("display",
"block");
            $('.paginated-checkout-navigation > :last-child').show();
        }
    });
});

function coresense_initialize_payment_processor() {

    braintree.client.create({
        authorization: $('#tokenization-key').val(),
    }, function (clientErr, clientInstance) {
```

```
        if (clientErr) {
            alert('Payment Processor Error [1]: '+clientErr.message);
            return;
        }

        braintree.dataCollector.create({
            client: clientInstance,
            kount: true
        }, function (err, dataCollectorInstance) {
            if (err) {
                console.log('Payment Processor Error [4]:
'+err.message);
                return;
            }

            $('#device_data').val(dataCollectorInstance.deviceData);
        });

        // Create a PayPal Checkout component
        if ($('#paypal_fragment').length) {
            braintree.paypalCheckout.create({
                client: clientInstance
            }, function (paypalCheckoutErr, paypalCheckoutInstance) {
                if (paypalCheckoutErr) {
                    console.error('Error creating PayPal Checkout:',
paypalCheckoutErr);
                    $(".paginated-checkout-controls >
a.right").css("display", "block");
                    $('.paginated-checkout-navigation > :last-
child').show();
                    return;
                }

                paypal.Button.render({
                    env: 'production',
                    commit: true, // This will add the transaction
amount to the PayPal button
                    /*payment: function () {
                        return paypalCheckoutInstance.createPayment({
                            flow: 'checkout', // Required
                            amount: {/literal}{$order-
>get_balance(true)}{literal}, // Required
                            currency: 'USD', // Required
                     });
                    },*/
                    payment: function () {
                        let balance_due = {/literal}{$order-
>get_due_balance_with_authorizations(true)}{literal};
                        $.ajax({
                            url:"ajax_targets/paypal_payment.php?
```

```
getOrderBalanceDue=true",
                            method: "GET",
                            async: false,
                            success: function(response) {
                                if(response.balance_due) {
                                    balance_due =
response.balance_due;
                                }
                            }
                        });

                        return paypalCheckoutInstance.createPayment({
                            flow: 'checkout', // Required
                            amount: balance_due, // Required
                            currency: 'USD' // Required
                        });
                    },
                    onAuthorize: function (data, actions) {
                        return
paypalCheckoutInstance.tokenizePayment(data, function (err, payload) {
                            $('#reference-card-
number').val(payload.nonce);
                            $(".paginated-checkout-controls >
a.right").css("display", "block");
                            $('.paginated-checkout-navigation > :last-
child').show();
                            $.ajax({
                                url:"ajax_targets/paypal_payment.php",
                                data: {
                                    payerID: data.payerID,
                                    paymentToken: data.paymentToken,
                                    paymentID: data.paymentID,
                                    referenceCardNumber: payload.nonce
                                },
                                method: "POST",
                                success: function(response) {
                                    window.alert('Payment Complete!');
                                }
                            });
                        });
                    },
                    onCancel: function (data) {
                        console.log('checkout.js payment cancelled',
JSON.stringify(data, 0, 2));
                        $(".paginated-checkout-controls >
a.right").css("display", "block");
                        $('.paginated-checkout-navigation > :last-
child').show();
                        $('#payment_types_select
option[value=""]').attr("selected",true);
```

```
                        $('#payment_types_select').val('').change();
                        $("#paypal_fragment").hide();
                        //$(".paginated-checkout-controls >
a.right").css("display", "none");
                        //$('.paginated-checkout-navigation > :last-
child').hide();
                    },
                    onError: function (err) {
                        console.error('checkout.js error', err);
                        //$(".paginated-checkout-controls >
a.right").css("display", "block");
                    }
                }, '#paypal_fragment').then(function () {
                    // The PayPal button will be rendered in an html
element with the id
                    // `paypal-button`. This function will be called
when the PayPal button
                    // is set up and ready to be used.
                });
            });
        }

        if ($('#card-number').length) {
            braintree.hostedFields.create({
                client: clientInstance,
                styles: {
                    'input': {
                        'font-size': '10px',
                    },
                    'input.invalid': {
                        'color': 'red'
                    },
                    'input.valid': {
                        'color': 'green'
                    },
                    'input': {
                        'height': '12px',
                    }
                },
                fields: {
                    number: {
                        selector: '#card-number'
                    },
                    cvv: {
                        selector: '#cvv'
                    },
                    expirationMonth: {
                        selector: '#expiration-month',
                        select: {
                            options: [
```

```
                                '01',
                                '02',
                                '03',
                                '04',
                                '05',
                                '06',
                                '07',
                                '08',
                                '09',
                                '10',
                                '11',
                                '12'
                            ]
                        }
                    },
                    expirationYear: {
                       selector: '#expiration-year',
                       select: true
                    }
                }
            }, function (hostedFieldsErr, hostedFieldsInstance) {
                if (hostedFieldsErr) {
                    alert('Payment Processor Error [2]:
'+hostedFieldsErr.message);
                    return;
                }

                document.querySelector('form#checkout_form
input[type="submit"]').addEventListener('click', function (event) {
                    if (($('#payment_ewallet_selector').length && $
('#payment_ewallet_selector').val() != '') || $('#reference-card-
number').val() != '') {
                       //return
document.getElementById('checkout_form').submit();
                    }

                    event.preventDefault();

                    hostedFieldsInstance.tokenize(function
(tokenizeErr, payload) {
                        if (tokenizeErr) {
                            alert('Payment Processor Error [3]:
'+tokenizeErr.message);
                            return;
                        }

                        var is_enable_3ds = $("#is_enable_3ds").val();
                        if(is_enable_3ds == 1){
                            braintree.threeDSecure.create({
                                client: clientInstance,
```

```
                                version: '2'
                            }, function (createError,
threeDSecureInstance) {
                              // set up lookup-complete listener
                                 // using Hosted Fields, use
`tokenize` to get back a credit card nonce


threeDSecureInstance.verifyCard({

amount: {/literal}{$order->get_balance(true)}{literal},
                                                                nonce:
payload.nonce, // Example: hostedFieldsTokenizationPayload.nonce
                                                                bin:
payload.binData, // Example:
hostedFieldsTokenizationPayload.details.bin
                                                                email:
$("#billing_contact_information_email").val(),

onLookupComplete: function (data, next) {

        // use 'data' here, then call 'next()'

        next();
                                                                },
                                    billingAddress: {

givenName: $("#shipping_contact_information_first_name").val(),

        surname: $("#billing_contact_information_last_name").val(),

        phoneNumber: $("#billing_contact_information_phone").val(),

        streetAddress: $
("#billing_contact_information_address_line_1").val(),

        locality: $("#billing_contact_information_city").val(),

        region: $("#billing_state_code").val(),

        postalCode: $
("#billing_contact_information_postal_code").val(),

        countryCodeAlpha2: $("#billing_country_code").val(),
                                                            },
                                    collectDeviceData: true,
                                    additionalInformation: {

workPhoneNumber: $("#billing_contact_information_phone").val(),
```

```
shippingGivenName: $
("#shipping_contact_information_first_name").val(),

shippingSurname: $("#shipping_contact_information_last_name").val(),

shippingAddress: {

streetAddress: $
("#shipping_contact_information_address_line_1").val(),

extendedAddress: $
("#shipping_contact_information_address_line_2").val(),

locality: $("#shipping_contact_information_city").val(),

region: $("#shipping_state_code").val(),

postalCode: $("#shipping_contact_information_postal_code").val(),

countryCodeAlpha2: $("#shipping_country_code").val(),
                                                    },
                                                    shippingPhone:
$("#shipping_contact_information_phone").val(),
                                                    productCode :
$("#productCode").val(),

acsWindowSize : $("#acsWindowSize").val()
                                                }
                                                    }, function
(err, response) {
                                                        if (err) {

alert('Payment Processor Error [3]: '+err);
                                    console.error('3D Secure
verification error:', err);
                                    // Handle error
                                    return;
                                }

                                console.log('3D Secure
verification response:', response);
                                var liabilityShifted =
response.liabilityShifted; // true || false
                                var liabilityShiftPossible =
response.liabilityShiftPossible; // true || false
                                if(liabilityShifted == true ||
liabilityShiftPossible == true) {
                                    $('#reference-card-
number').val(payload.nonce);
                                    return
```

```
document.getElementById('checkout_form').submit();
                                        }
                                    });

                                });

                            }


                           $('#reference-card-number').val(payload.nonce);
                           return
document.getElementById('checkout_form').submit();

                        });
                    }, false);
                });
            }
        });
}


</script>
{/literal}
```