verifone

https://verifone.cloud/docs/uk-gateway/inject_tokenization
Updated: 29-Oct-2021

Tokenization

To tokenize cardholder data through Inject follow these steps:

1. A payment page is presented to the customer, including a payment form rendered by jsclient.
2. The customer inputs their card details, then submits the form.
3. jsclient transmits the card details to our server, which generates a tokenized version of the card details, returning this to jsclient.
4. jsclient posts the card token, along with any custom form data, to the server endpoint specified in your payment form.
5. Initiate the card transaction from your server as described here.

Create the payment form

To begin, you will need to create a web page that displays a form where the card details will be entered by the customer and submitted.

- The `<form>` element must be tagged with a unique `id` attribute specified by you, which will be used by jsclient to find the form and append the required card details fields and submit button after any custom fields that you choose to define.
- The form's `method` attribute must be set to `POST`. The `GET` method is not supported by jsclient.
- The form's `action` attribute must point to the URL on your server where you intend to receive the card token and initiate the transaction.

```
<form id="my_payment_form" method="POST" action="/initiate_transaction">
...
</form>
```

Your web page must include the jsclient script loaded from our server, then call the method `jsclient.injectForm(HTMLElement id)` with the `id` value of the `<form>` element you tagged. In order to transfer the sensitive card details to our server, the jsclient script also requires a valid `organisation_id` from the platform. As a result, the card token created will be visible in the scope of the provided organisation.

```
<script src="https://sandbox.omni.verifone.cloud/static/jsclient/script.js"
></script>
...
<script>
    jsclient.injectForm(document.getElementById('my_payment_form'), {
        "organisation_id": "58a60e6e27c7ae173e332853"
```

```
    });
</script>
```

Here is an example of a complete web page for card payments, including the default bootstrap stylesheet, and a custom hidden form field named `my_order_id`.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>My payment page</title>
<link rel="stylesheet" href=
"//maxcdn.bootstrapcdn.com/bootswatch/3.2.0/sandstone/bootstrap.min.css" />
<script src="https://sandbox.omni.verifone.cloud/static/jsclient/script.js"
></script>
</head>

<body>
<form id="my_payment_form" method="POST" action="/initiate_transaction">
<input type="hidden" name="my_order_id" value="12345" />
<script>
   jsclient.injectForm(document.getElementById('my_payment_form'), {
                "organisation_id": "58a60e6e27c7ae173e332853"
            });
</script>
</form>
</body>

</html>
```

If you now load your page in a web browser, it should display at least the following items.

- card number field
- expiry date field
- security code field
- submit button

Learn more about jsclient

This is a simplistic example, for clarity. In practice, you will likely want to customize your form, and the behaviour of jsclient. More information about jsclient can be found here, including its configuration options and styling capabilities.

Receive the card token

Now that your payment form is ready, you need to implement code that will receive the card token as part of the form submission, then initiate the transaction by posting the required data to our server.

When the customer clicks the submit button, jsclient causes their card details to be securely transmitted directly to our server, where they are stored and tokenized, with the card token being returned to jsclient. Then jsclient allows the form submission to proceed, communicating the card token to your server as part of the form

submission.

For example, if your form's `method="POST"` and its `action="/initiate_transaction"`, then you should create an endpoint that listens for an HTTPS POST request at `/initiate_transaction` that contains a URL-encoded request body that looks something like this, representing the token that we generated in exchange for the card details that the customer submitted.

```
card=563ff2c8f0103bb04be177ab
```

The values of any custom fields that you specified will also be included in the POST data, along with the card token.

Inject is a JavaScript client module that handles the rendering of web form fields and a submit button used to capture the customer's card details. The module can be found at `$BASEURL/static/jsclient/script.js`.

The Inject module can be used to either tokenize cardholder data or initiate Alternative Payment Method (APM) payments like Google Pay. Read more on what you need to factor in when using Inject over other integration methods here and read more here to see how this influences your PCI compliancy requirements.

Initiate the transaction

After receiving the card token proceed here to initiate the transaction.