Monetra® Payment Software

PaymentFrame Guide

PaymentFrame Guide

Main Street Softworks, Inc.

Revision: 1.2

Publication date March 28, 2018
Copyright © 2018 Main Street Softworks, Inc.

Legal Notice

Table of Contents

1 Revision History

Version	Date	Changes
v1.0	2017-08-30	• Initial document.
v1.1	2017-10-31	• Documented the ability to split cardholder name field into first and last.
v1.2	2018-02-12	• This update applies to the "HMAC Parameters" section. It enhances the description for the 'hmac-css-url' parameter and adds the three new ones below: * `hmac-include-cardholdername` * `hmac-include-street` * `hmac-include-zip`

2 Monetra PaymentFrame

2.1 Introduction

PaymentFrame is a Monetra feature that allows you to embed a secure iframe containing a payment form on an HTTPS-enabled web page. This prevents credit card data from ever touching your systems, while allowing your customers to complete seamless ecommerce transactions on your site.

> PCI Notice: This approach to eCommerce integration follows the PCI Council's best practices as outlined in https://www.pcisecuritystandards.org/pdfs/best_practices_securing_ecommerce.pdf.

2.2 How It Works

> Note: In order to use the PaymentFrame, you will need credentials (username and password) for a merchant subuser on the Monetra (or TranSafe) instance you are connecting to. This subuser must have 'obscure sensitive information' enabled.

To make the PaymentFrame as easy as possible to use, we provide a support package available for download at https://www.monetra.com/developers. This download includes a Javascript file, `MonetraPaymentForm.js`, which defines a utility class called `MonetraPaymentForm`. You can use this class in your own Javascript to handle the process of requesting the payment form iframe from Monetra. The download also includes an example Javascript file that demonstrates how to use the `MonetraPaymentForm` class, as well as multiple example CSS files for styling the form.

1. *Prepare your credentials:* Before rendering your payment page (the page that will host the iframe), you will need to generate a 256-bit keyed-hash message authentication code (HMAC-SHA256) See HMAC parameters below.

2. *Add iframe element and components:* Your html payment page must include an empty iframe element (with no `src` attribute specified) where you would like the payment form to appear. The HMAC message components (along with the HMAC itself from step 1) must be included as data- attributes on the iframe element. Note: The order of the data attributes in the opening iframe tag must match the order in which they are concatenated into the HMAC message. The ordering is crucial for Monetra/TranSafe to be able to properly verify the HMAC. See HMAC parameters below.

3. *Load script:* Your payment page will need to load `MonetraPaymentForm.js`, a Javascript file that takes care of the bulk of the work of requesting the payment iframe.

4. *Load Payment Form:* From here you'll just need to write a few lines of your own Javascript to instantiate a `MonetraPaymentForm` object and request the iframe. You will need to define a callback function to be executed after the user's payment information has been submitted. The full sequence would be instantiate `MonetraPaymentForm`, define callback, request iframe.

5. *Submit Cardholder Data:* Once the iframe has loaded, it will render a payment form into which the user can enter their payment information. When the user submits the form, their information will be sent directly to the Monetra/TranSafe server, which will return a CardShield ticket.

6. *Process response:* At this point, the callback function that you defined in your Javascript will be executed, receiving a JSON object representing the Monetra/TranSafe server's response (including the CardShield ticket) as a parameter.

7. *Process the ticket:* Once you have a valid ticket then you can communicate directly with the Monetra/TranSafe system to run a standard transaction such as as SALE or a PRE-AUTH as defined in the Monetra Application Interface Guide.

2.3 HMAC Parameters

Legend:
Req = Required or not [Y=Yes C=Conditional O=Optional]

Key	Req	Description
`hmac-hmacsha256`	Y	Generated HMAC-SHA256 using the merchant's password as the key. The message must be a concatenation (with no delimiters) of all of the other hmac- values you are providing, in the same order in which they are specified as data attributes on the `iframe` element.
`hmac-timestamp`	Y	Standard Unix timestamp. Must be within 15 minutes of server's time.
`hmac-domain`	Y	The domain of the page embedding the iframe. This must match exactly what would show in the browser URL bar, including 'https://' (and port, if it is explicitly set).
`hmac-sequence`	Y	Merchant-specified sequence number. May be alphanumeric.
`hmac-username`	Y	Username used for authentication by the Monetra server. This must be a subuser with 'obscure sensitive information' enabled.
`hmac-css-url`	O	URL of CSS file the iframe should load. It must be served from the same domain as `hmac-domain`. If no CSS URL is provided, the payment form will be rendered with default styling (see "Styling the Payment Form" for more information).
`hmac-include-cardholdername`	O	Indicates whether the cardholder name field should be present on the payment form. Value can be "yes" or "no". Defaults to "yes" if not provided. Only provide "no" here if cardholder name is collected elsewhere during your checkout process, and sent along with the ticket in the Monetra sale request.

`hmac-include-street`	O	Indicates whether the street address field should be present on the payment form. Value can be "yes" or "no". Defaults to "yes" if not provided. Only provide "no" here if street address is collected elsewhere during your checkout process, and sent along with the ticket in the Monetra sale request.
`hmac-include-zip`	O	Indicates whether the zip code field should be present on the payment form. Value can be "yes" or "no". Defaults to "yes" if not provided. Only provide "no" here if zip code is collected elsewhere during your checkout process, and sent along with the ticket in the Monetra sale request.
`hmac-expdate-format`	O	Format of expiration date the iframe should use. Choices: * single-text: Free form text entry. Default if not specified. * separate-selects: Two drop downs for month and year. * coupled-selects: Two drop downs for month and year in a container.
`hmac-cardholdername-format`	O	Single, or split cardholder name. Choices: * `separate-firstlast` * `combined-firstlast` `separate-firstlast` will split the card holder name field into first and last name fields. This field must be part of the hmac if present. It comes after 'expdate-format' in the hmac data.

Note: Response Parameters are the same as returned by the Monetra POST Protocol ticket request. The hmac- timestamp and hmac-sequence will be used as part of the POST request to allow the hmac-hmacsha256 response from the ticket request to be validated.

2.4 Styling the Payment Form

The `iframe` content can be styled using custom CSS. You can use the classes and IDs documented below in your CSS to style the payment form and the elements it contains.

You can also provide custom text for the form's labels by specifying appropriate CSS rules. Each label contains an empty span element, so that you can use the ::before selector in conjunction with the content property to insert your desired text into the label. For example, in order to provide custom text for the ZIP code label, you would use a CSS rule similar to the following:

```css
#payment-zip-label span::before {
    content: "Custom text here";
}
```

Note: If you do not provide custom text for any label, it will default to the value referenced in the "IDs" table below.

CLASSES	
Class	Description

`payment-form-label`	Applied to all label elements in the payment form. Each of these elements contains an input or select element into which payment data will be entered.
`payment-expdate-label`	If hmac-expdate-format is set to separate-selects or coupled-selects, this will be applied to the two label elements that contain the select drop-downs for expiration date month and year.
`payment-cardholdername-label`	If hmac-cardholdername-format is set to separate-firstlast, this will be applied to the two label elements that contain the first and last name text entry fields.

ID's	
ID	Description
`monetra-payment-form`	The payment `form` element.
`payment-account-label`	The label containing the account (credit card number) input element. Label text defaults to 'Card Number'.
`payment-expdate-container`	If hmac-expdate-format is set to coupled-selects, this is the div that contains the two select elements.
`payment-expmonth-label`	If hmac-expdate-format is set to separate-selects or coupled-selects, this is the label that contains the expiration month select element. Label text defaults to 'Expiration Month'.
`payment-expyear-label`	If hmac-expdate-format is set to separate-selects or coupled-selects, this is the label that contains the expiration year select element. Label text defaults to 'Expiration Year'.
`payment-expdate-label`	If hmac-expdate-format is set to single-text, this is the label that contains the 'expdate' (expiration date) input element. Label text defaults to 'Expiration Date'.
`payment-cardholdername-label`	The label containing the `'cardholdername'` input element. Label text defaults to 'Cardholder Name'.
`payment-cardholdernamefirst-label`	If hmac-cardholdername-format is set to separate-firstlast, this is the label that contains the first name field. Label text defaults to Cardholder First Name.
`payment-cardholdernamelast-label`	If hmac-cardholdername-format is set to separate-firstlast, this is the label that contains the last name field. Label text defaults to Cardholder Last Name.
`payment-street-label`	The label containing the 'street' (street address) input element. Label text defaults to 'Street Address'.
`payment-zip-label`	The label containing the 'zip' (ZIP code) input element. Label text defaults to 'ZIP Code'.
`payment-cv-label`	The label containing the 'cv' (card verification value) input element. Label text defaults to 'CV'.

`payment-submit-button`	The button element used to submit the form.

2.5 Code Example (PHP)

```php
1  <?php
2
3  /* Values that will be needed for generating the HMAC */
4  $host_domain = "https://your.website.com";
5  $monetra_username = "test_retail:public";
6  $monetra_password = "publ1ct3st";
7
8  $hmac_fields = [];
9
10 /* "timestamp", "domain", "sequence", and "username" are the required HMAC fields. */
11
12 /* Current Unix timestamp */
13 $hmac_fields["timestamp"] = time();
14
15 /* Domain of the website that will host the iframe */
16 $hmac_fields["domain"] = $host_domain;
17
18 /* Merchant-specified sequence value for tracking/verification purposes.
19  * In production this should be dynamically generated.
20  */
21 $hmac_fields["sequence"] = "abc123";
22
23 /* Username of the Monetra merchant user that will be used to request the iframe
24  * and generate the ticket
25  */
26 $hmac_fields["username"] = $monetra_username;
27
28 /* Optional field. This is the URL of the CSS file that will be used to style
29  * the iframe's contents.
30  */
31 $hmac_fields["css-url"] = $host_domain . "/css/iframe.css";
32
33 /* Optional field. This will direct Monetra to generate the form with separate
34  * select elements for the expiration date month and year, rather using than a
35  * single text element for the expiration date.
36  */
37 $hmac_fields["expdate-format"] = "separate-selects";
38
39 /* Concatenate all of the defined HMAC fields into a string with no delimiters */
40 $data_to_hash = implode("", $hmac_fields);
41
42 /* Generate the HMAC, using the Monetra merchant user's password as the key */
43 $hmac = hash_hmac('sha256', $data_to_hash, $monetra_password);
44
45 /* Assemble a string containing the "data-" attributes for the iframe element.
46  * This will consist of the HMAC itself and all of the fields included in the HMAC.
47  */
48 $iframe_attributes = [
49   'data-hmac-hmacsha256="' . $hmac . '"'
50 ];
51 foreach ($hmac_fields as $key => $value) {
```

```
52    $iframe_attributes[] = 'data-hmac-' . $key . '="' . $value . '"';
53  }
54  $iframe_attribute_string = implode(" ", $iframe_attributes);
55
56  /* Render the payment page HTML. */
57  ?>
58
59  <DOCTYPE html>
60  <html>
61   <head>
62    <meta name="viewport" content="width=device-width, initial-scale=1">
63    <title>Example Shopping Site</title>
64    <link rel="stylesheet" type="text/css" href="./css/host.css" />
65   </head>
66   <body>
67    <main>
68     <h1>Checkout Page</h1>
69     <p>
70      Please fill out your payment information below.
71     </p>
72     <iframe id="iframe" <?php echo $iframe_attribute_string; ?>></iframe>
73    </main>
74    <script src="./js/MonetraPaymentForm.js"></script>
75    <script src="./js/host.js"></script>
76   </body>
77  </html>
78
```