Paythru Remote Fields

Paythru Remote Fields are an alternative integration method for the Paythru Client POST API. The integration consists of contructing a basic javascript configuration object and including a javascript library from Paythru's servers. The script will add iFrames to a checkout form to collect sensitive card data. Using iFrames in this way can help to reduce your PCI requirements.

How it works

In order to be eligible for PCI – SAQ A compliance, all processing of cardholder data must be entirely outsourced to PCI DSS validated third-party service providers. To support this goal the remote fields javascript inserts iFrames into your form at runtime. The iFrames contain the fields that collect card data, the fields themselves are hosted within Paythru's fully PCI level 1 compliant environment.

Most attributes of the input fields to be hosted by Paythru can be specified in the `fields` parameter. This allows you to maintain control over the look and feel of the page.

When the form is submitted, the javascript interrupts the default action. Each individual iFrame then submits its data to Paythru. If there are any errors, a relevant error message is appended after each field. The error message will be placed within a paragraph tag with a class 'paythru-field-error'. The look and feel of the error message can be customised by defining CSS for this class. You can further affect the display of errors by using the optional `errorMessages` JSON object to define the text you'd like to display for each field error, or override the error handling completely by defining your own `errorHandler`.

If the form passes validation the data which was collected is sent to the `nextPageUrl` via HTTP POST, with the sensitive data masked. Further actions such as payment processing and card tokenisation may be performed using the session key. Please refer to Paythru's Gateway API and Card Tokenisation API for more details.

Include the JS library

The javascript is available from https://ws.paythru.com/remotefields/v3/remotefields.min.js. There is no need to switch which js you use between live and test environments.

```
<script type="text/javascript"
src="https://ws.paythru.com/remotefields/v3/remotefields.min.js"></script>
```

Setup Parameters

The following parameters are available for the `remotefields.setup` function call which describes how the form should be built and behave.

Key	Type/Possible values	Required	Default value	Description
sessionKey	String	Y		The session key generated using the initiate function of the client POST API. See http://www.paythru.com/developers/client_post_api_v3.php
env	"test" or "live"	N	"live"	Your current environment.
formId	String	Y		The value of the *id* attribute of the card form. The sensitive card fields of this form will be replaced with iFrames
nextPageUrl	String	Y		The URL to the next page. Once the form is submitted and validation passes, an associative array of data from the form will be passed to the *nextPageUrl* via HTTP POST method.
errorMessages	JSON object	N		This option can be used to set custom error messages. The object should contain key-value pairs with the key as the error code and value as the custom error messages. For the list of possible errors, see http://www.paythru.com/developers/client_post_api_v3.php
errorHandler	Function(data)	N		The callback function that is executed if there is an error in the submitted form. The data will be a JSON object which contains the reason for the form validation failure.
fields	JSON Object	Y		This option should be used to define the input fields. See the section about Fields Parameters for more detail.

Fields Parameters

The setup function accepts a parameter called fields, which is a JSON object describing each of the sensitive data entry fields in your form (identified by formId). The following table lists the parameters it can support to specify the style and behaviours of individual fields.

It should contain a list of objects with the key as the id of an HTML element and the value as an object with all the attributes of the input element to be generated.

Key	Type	Required	Description
id	String	Y	As per W3C html specification
name	String	Y	The name for the field. Must match the appropriate field name from the Paythru Client POST API
type	String	Y	Accepts all valid values for type attribute for input tag. For dropdown, set the type to *select*
placeholder	String	N	As per W3C html specification
frameHeight	String	N	Height of the iFrame. Include the unit type (i.e. pixels, %)

frameWidth	String	N	Width of the iFrame. Include the unit type (i.e. pixels, %)
style	String	N	The style attribute. Only specific styles are supported due to security concerns. Accepted style attributes are listed in the Allowed style attributes section of this document.
styleOnFocus	String	N	Accepts the same data format as the above `style` attribute and applies the :focus pseudoclass.
options	String	N	If 'type' is set to 'select', the options should be a key-value pair that will be converted to the options of the dropdown input
value	String	N	If your custom form validation fails, use this field to pre-populate the field with the masked data
align	String	N	As per W3C html specification
alt	String	N	As per W3C html specification
checked	String	N	As per W3C html specification
disabled	String	N	As per W3C html specification
height	Integer	N	As per W3C html specification
max	Integer	N	As per W3C html specification
maxlength	Integer	N	As per W3C html specification
min	Integer	N	As per W3C html specification
pattern	String	N	As per W3C html specification
required	String	N	As per W3C html specification
size	Integer	N	As per W3C html specification
width	Integer	N	As per W3C html specification

Allowed style attributes

All of the allowed rules will also be supported if prefixed with any one of:

`-webkit-, -moz-, -ms-, -o-`

Allowed rules

The following rules will be allowed (full string match):

`color, height, max-height, box-shadow, height, max-height, max-width, min-height, min-width, width, letter-spacing, text-align, text-transform, text-decoration, background-color`

Allowed wildcard rule matches

The following wildcard matches will be allowed (all rules matching):

`border* (except border-image) margin* padding* font* (except font-face)`

Rejected values

The following strings will be rejected if found within the value of a style attribute:

`url, script, http, expression, javascript`

Sample Code

HTML Code

```
<form method="post" id="card-form">
    <h3>Your card details</h3>
    <div class="form-group">
        <label for="cardNumber" class="col-md-4 control-label">Card
number*</label>
        <div class="col-md-8" id="card-number"></div>
    </div>
    <div class="form-group">
        <label for="cardExpiryDate" class="col-md-4 control-label">Expiry
date*</label>
        <div class="col-md-4" id="card-expiry-month"></div>
        <div class="col-md-4" id="card-expiry-year"></div>
    </div>
    <div class="form-group">
        <label for="cardCV2" class="col-md-4 control-label">Security
Code*</label>
        <div class="col-md-8">
            <div id="card-cvv"></div>
            <div>These are the 3 digits on the back of your card</div>
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-5">
            <input type="checkbox" name="saveCard" value="1" /> Save Card
        </div>
    </div>
    <div class="btn-row">
        <input type="submit" name="submit" class="btn btn-primary pull-right"
value="Next" />
    </div>
</form>
```

Javascript Code

```
<noscript><h1>Your browser does not support JavaScript.</h1></noscript>
<script type="text/javascript"
src="https://ws.paythru.com/remotefields/v3/remotefields.min.js"></script>
<script type="text/javascript">
remotefields.setup({
    'env': 'test',
    'sessionKey': '44b9819395f########92ca92ad3a97f',
    'formId': 'card-form',
    'nextPageUrl': ''https://your-website/saveMaskedCard'',
    'fields': {
        'card-number': {
            'id': 'cardNumber',
            'name': 'cardNumber',
            'type': 'tel',
            'placeholder': '4444 3333 2222 1111',
            'frameHeight': '38',
            'frameWidth': '420',
            'style': 'border-radius: 4px;font-size: 16px; height: 34px; padding:
6px 12px;border: 1px solid #ccc;width: 100%;'
        },
        'card-expiry-month': {
            'name': 'cardExpiryMonth',
            'type': 'select',
            'frameHeight': '38',
            'frameWidth': '200',
            'options': {
                1: 'Jan', 2: 'Feb', 3: 'Mar', 4: 'Apr', 5: 'May', 6: 'Jun',
                7: 'Jul', 8: 'Aug', 9: 'Sept', 10: 'Oct', 11: 'Nov', 12: 'Dec',
            },
            'style': 'border-radius: 4px;font-size: 16px; height: 34px; padding:
6px 12px;border: 1px solid #ccc;width: 100%;'
        },
        'card-expiry-year': {
            'name': 'cardExpiryYear',
            'type': 'select',
            'frameHeight': '38',
            'frameWidth': '200',
```

```
            'options': {'2016' : '2016', '2017' : '2017', '2018' : '2018'},
            'style' : 'border-radius: 4px;font-size: 16px; height: 34px;
padding: 6px 12px;border: 1px solid #ccc;width: 100%;'
        },
        'card-cvv': {
            'id': 'cardCV2',
            'name': 'cardCV2',
            'type': 'tel',
            'frameHeight': '38',
            'frameWidth': '100',
            'style': 'border-radius: 4px;font-size: 16px; height: 34px; padding:
6px 12px;border: 1px solid #ccc;width: 100%;'
        }
    }
});
</script>
```

Customer errorHandler example

```
<script type="text/javascript">
remotefields.setup({
    'sessionKey': '44b9819395f########92ca92ad3a97f',
    'errorHandler': 'myErrorHandler',
    'nextPageUrl': 'https://your-website/saveMaskedCard',
    'fields': {
        // Define fields here
    }
});

/**
Below is an example of the JSON object received by the error handler
function.

{
    "cardNumber":"444433332222111",
    "cardExpiryDate":"1119",
    "cardCV2":"",
    "cardholderName":"John Smith",
    "errors":{
        "cardNumber":{
            "errorCode":"108",
            "traceCode":"010727108",
            "errorMessage":"Invalid card number"
        },
        "cardCV2":{
            "errorCode":"164",
            "traceCode":"010727164",
            "errorMessage":"CVV cannot be less than 3 digits"
        }
    }
}

// Here, the error message for cardNumber is added inside a div with the
class ‘error-cardNumber’
*/

function myErrorHandler (returnObj) {
    document.getElementById('error-heading').html('Unable to save your card
details, please correct the errors below;');

    var errorPrefix = '<ul class="form__error"><li>';
    var errorSuffix = '</li></ul>';

    Object.keys(resultJson.errors).forEach(function(key) {

        document.getElementById('error-'+key).html(errorPrefix +
resultJson.errors[key].errorMessage + errorSuffix);

    });
}</script>
```