

Payeezy.JS Integration Guide

July 2015

Contents

Quickly integrate your applications with Payeezy JS	3
Minimum technical requirements	3
Payeezy.JS initialization.....	3
Request payload for Payeezy JS GET – Generate token method	11
Downloadable information	11
Submitting/Generating transaction (method of payments) with example.....	11
Additional capabilities API	12

Quickly integrate your applications with Payeezy JS

If you want to enable secure and convenient payments in your payment applications, this guide will get you up and running quickly. Payeezy handles all the heavy lifting of the complex tokenization and protects your customers' transactions. It is simple to create a developer test account and apply for a merchant account through our developer portal.

PayeezySM, from First Data, is part of the Small Business Solutions Suite, which includes CloverTM, InsighticsSM, PerkaTM and TransArmor[®]. The Payeezy eCommerce Solution empowers SMBs to expand their horizons and easily grow their business online or via mobile by reaching new customers no matter where they are.

This document refers to the Payeezy.JS integration method included within the overall Payeezy eCommerce Solution. Henceforth, all references to Payeezy are in relation to Payeezy APIs.

Minimum technical requirements

1. TransArmor must be enabled on your merchant account to do token based transactions. Contact your merchant account representative or call 1-855-448-3493 for more details on how to enable this.
2. Compatible browsers:

Browser	Version
Firefox	v33.0
Chrome	v39.x
Internet Explorer	IE6+
Safari	v5.1.7

3. Sign up for a test account and begin the integration process!

Payeezy.JS initialization

Step by step integration

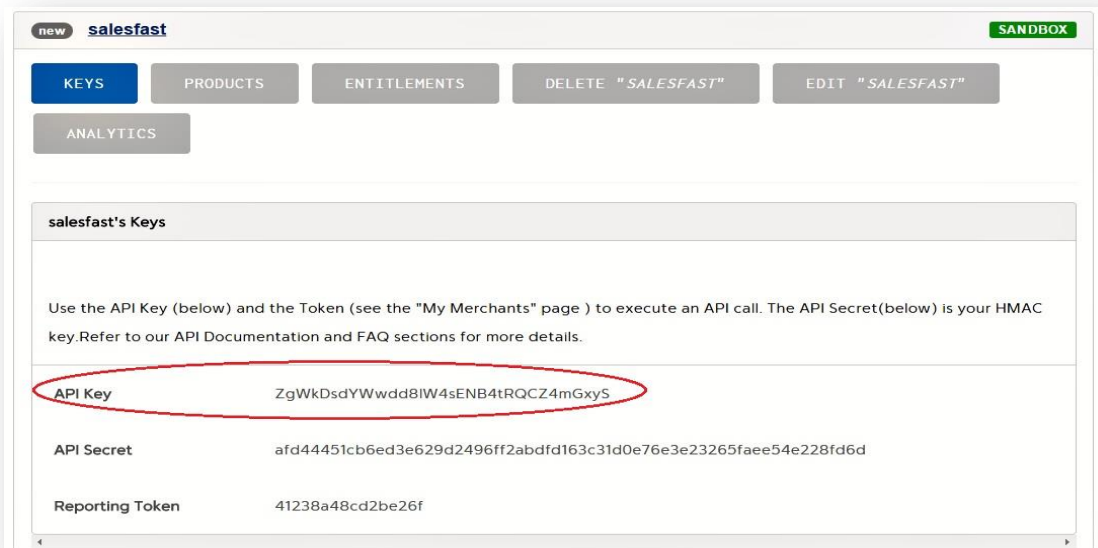
- a. Look at the the steps in prerequisites to register your developer account and create your API. Link for prerequisites https://github.com/payeezy/get_started_with_payeezy/raw/master/get_started_with_payeezy042015.pdf
- b. Developer portal setup:
Provision your developer account on developer.payeezy.com and create a sandbox API for Payeezy.JS.
- c. You will need the following parameters to initialize Payeezy JS.

Version	Name of the JS file	Release date	Supported Token Types	apikey Required	Entitlements Required	TA_TOKEN Required	JS_Security_Key Required
Payeezy JS - v3	payeezy_v3.2.js	1-Jul-15	FDTOKEN	Yes	No	Yes	Yes

Let's review how to capture them.

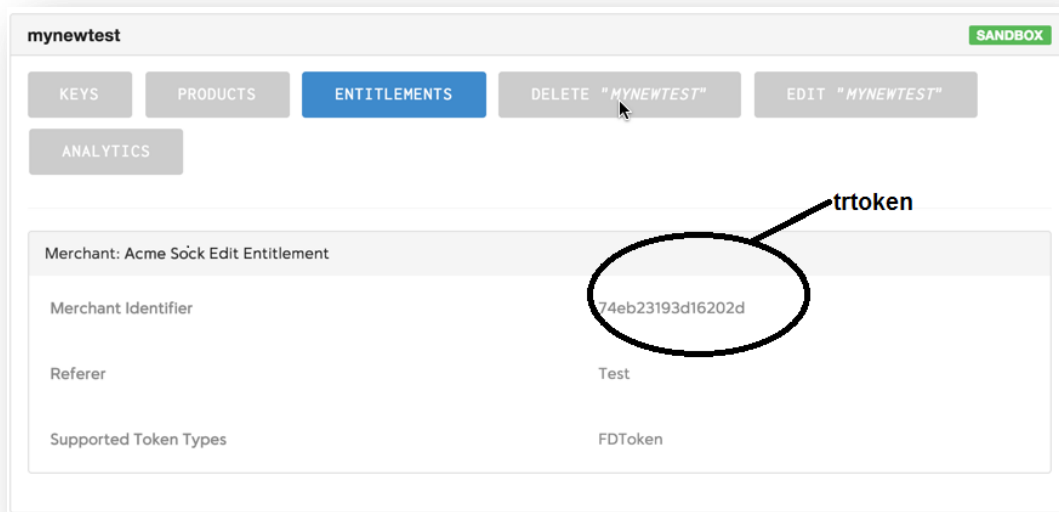
1. On the developer portal, capture the API Key you have created in Step #2.

[developer.payeezy.com webpage: 1](#)



2. For the existing *APIs*, entitlements will show on the screen with 'Entitlements' tab and details, as shown below {If the entitlements were set before}

[developer.payeezy.com webpage: 2 \(existing entitlements - Readonly\)](#)



For new APIs, the Entitlements tab will not be displayed. Entitlements will be assigned and managed by developer.payeezy.com internally.

developer.payeezy.com webpage: 3 ([Newly registered apis after June 30,2015](#))

new onemoretest

SANDBOX

KEYS

PRODUCTS

DELETE "ONEMORETEST"

EDIT "ONEMORETEST"

ANALYTICS

onemoretest's Keys

Use the API Key (below) and the Token (see the "My Merchants" page) to execute an API call. The API Secret(below) is your HMAC key.Refer to our API Documentation and FAQ sections for more details.

API Key	XZOTimRgBEjNB3XRFSRIhHdy9PGA38pR
API Secret	4d8839fb2ede230e981e176e31f066e8eaa68645aad37a7dd6ebb07ec31145b
Reporting Token	1a97b29be15480d4

3. Follow below steps to retrieve and capture **JS Security Key** parameter from developer.payeezy.com
 - a. Login to developer portal
 - b. Click on My Merchants
 - c. You will find JS security Key under **SANDBOX** table

My Merchants

SANDBOX

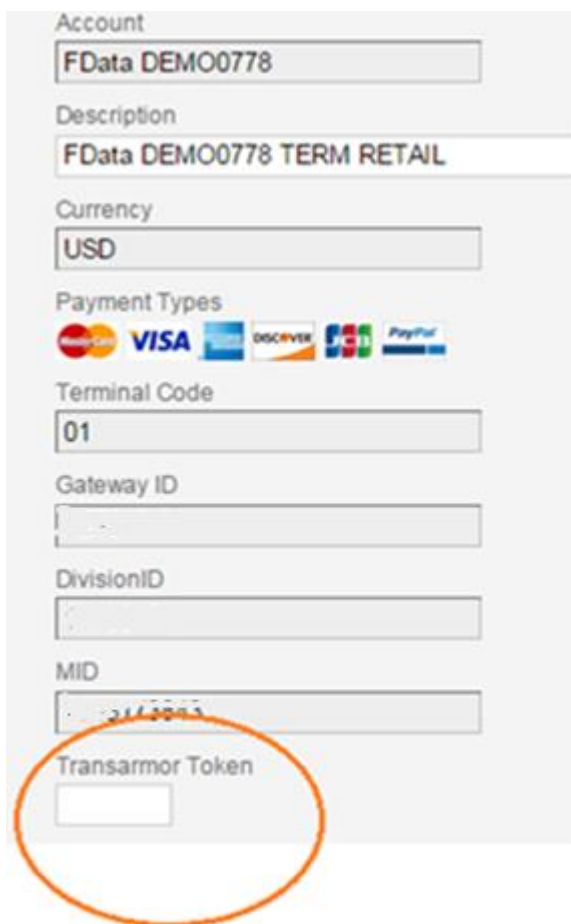
LIVE

What's this ?

Merchant ID	Merchant Name	CSR	App Label	Token	JS Security Key	View Details	Status
993216554887	Acme Sock	Not Applicable	merchant.com.cert.PAACBO	fdoa-a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6	js-d540fbd59173ae1a7073515180df01fed540fbd59173ae1a	Not Applicable	Initiated

4. Follow below steps to retrieve and capture **ta_token** parameter from Payeezy Gateway for LIVE merchant account.
 - a. Please log into your Payeezy Gateway virtual terminal (<https://globalgateway4.firstdata.com>)
 - b. Navigate to Terminals tab and select your terminal.
 - c. As shown below, retrieve the transarmor token value. This is your ta_token parameter.
 - d. **Please Note:**
 - For SANDBOX (CERT) test merchant account, please set **ta_token=NOIW**.
 - For LIVE merchant account, enable transarmor and capture tatoken value as shown below.

[Globalgateway4.firstdata.com](https://globalgateway4.firstdata.com) webpage: 1



5. **auth** : Possible values are true or false.
 - a. **false** – gets you a FDTOKEN without 0\$ auth. The token generated is valid for “authorize”, “purchase” and reversals (“capture”, “void” or “refund/settled”) transactions. ta_token parameter is mandatory for auth=false. **Note:** cvv check will not happen here and card information will not be validated.
 - b. **true** – gets you a FDTOKEN with 0\$ auth. The token generated is valid for “authorize” transaction only.

6. Initializing the Payeezy.js :

Payeezy JS implementation can be initialized by placing the <script> tag inside the <head> tag of the Payment page. Download copy of payeezy.js from github. Add link to JQuery reference.

```
<script src="/payeezy_v3.2.js" type="text/javascript"></script> <script
src="http://code.jquery.com/jquery-1.11.3.min.js" type="text/javascript"></script>
```

7. Initialize Payeezy.js by the apiKey and the Merchant Identifier inside the <script> tag as shown below. For every merchant, api key and merchant identifier will be unique and the same can be retrieved from the developer portal (sample values shown below)

```
<script>
Payeezy.setApiKey(apiKey);
Payeezy.setJs_Security_Key(js_security_key);
Payeezy.setTa_token(ta_token);
</script>
```

8. Payment Information Set Up:

- c. Payeezy.js has to be initialized for setting up the Payment information page. Please look at the highlighted attribute names below that are required to generate the payment token. All the fields that are not PII will be submitted to the Merchant server.

```
<form method="post" name="payment-info-form" id="payment-info-form">
  <table>
    <tr>
      <td align="right">API Key :</td>
      <td><input type="text" name="apikey" id="apikey"
        value="y6pWAJNyJyjGv66IsVuWnklkKUPFbb0a" /></td>
    </tr>
    <tr>
      <td align="right">API Secret :</td>
      <td><input type="text" name="apisecret" id="apisecret"
        value="86fbae7030253af3cd15faef2a1f4b67353e41fb6799f576b5093ae52901e6f7" />
      </td>
    </tr>
    <tr>
      <td align="right">JS Security Key :</td>
      <td><input type="text" name="js_security_key" id="js_security_key"
        value="js-6125e57ce5c46e10087a545b9e9d7354c23e1a1670d9e9c7" />
      </td>
    </tr>
  </table>
</form>
```

Code continues....)

```
<tr>
  <td align="right">Merchant Token :</td>
  <td><input type="text" name="token" id="token"
    value="fdoa-a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6" /></td>
</tr>
<tr>
  <td align="right">Auth :</td>
  <td>
    <select name="auth" id="auth" payeezy-data="auth">
      <option value="false">>false</option>
      <option value="true">>true</option>
    </select>
  </td>
</tr>
<tr>
  <td align="right">TA Token :</td>
  <td><input type="text" name="ta_token" id="ta_token"
    payeezy-data="ta_token" value="NOIW" /></td>
</tr>
<tr>
  <td align="right">Card Type :</td>
  <td>
    <select payeezy-data="card_type">
      <option value="visa">Visa</option>
      <option value="mastercard">Master Card</option>
      <option value="American Express">American Express</option>
      <option value="discover">Discover</option>
    </select>
  </td>
</tr>
<tr>
  <td align="right">Cardholder Name :</td>
  <td><input type="text" payeezy-data="cardholder_name"
    value="xyz" /></td>
</tr>
<tr>
  <td align="right">Card Number :</td>
  <td><input type="text" payeezy-data="cc_number"
    value="4788250000028291" /></td>
</tr>
<tr>
  <td align="right">CVV Code :</td>
  <td><input type="text" payeezy-data="cvv_code" value="123" />
  </td>
</tr>
<tr>
  <td align="right">Expiry Date :</td>
  <td>
    <select payeezy-data="exp_month">
      <option value="01">01</option>
      <option value="10">10</option>
      <option value="11">11</option>
      <option value="12" selected>12</option>
    </select>
    <select payeezy-data="exp_year">
      <option value="15">2015</option>
      <option value="16">2016</option>
      <option value="17">2017</option>
    </select>
  </td>
</tr>
```


Code continues....)

```
</table>
<div id="generate-token">
  <table>
    <tr>
      <td></td>
      <td><button type="submit">Generate token</button></td>
    </tr>
  </table>
</div>
</form>
```

9. **Creating the Payment Token:** Below code snippet can be used to capture the submit event and then create the payment token using the Payeezy.JS implementation. It's not required to use jQuery, the same functionality can be accomplished with native javascript.

```
jQuery(function($) {
    $('#payment-info-form').submit(function(e) {
        var $form = $(this);
        var apiKey = document.getElementById("apikey").value;
        var js_security_key =
document.getElementById("js_security_key").value;

        <!-- Setting Key parameters for Payeezy gettoken method -->
        Payeezy.setApiKey(apiKey);
        Payeezy.setJs_Security_Key(js_security_key);
        Payeezy.setTa_token(ta_token);
        Payeezy.createToken(responseHandler);
        $('#someHiddenDiv').show();
        return false;
    });
});
```

“responseHandler” is a merchant defined JS function, which will be called when the creation of payment token is successful. Example of this function is provided in the next section.

10. **Sending the Payment Information to Merchant Server:**

“responseHandler” method takes two arguments status and response(a javascript object). If the payment token

```
var responseHandler = function(status, response) {
    var $form = $('#payment-info-form');
    $('#someHiddenDiv').hide();
    if (status != 201) {
        if (response.error && status != 400) {
            var error = response["error"];
            var errormsg = error["messages"];
            var errorcode = JSON.stringify(errormsg[0].code, null, 4);
            var errorMessagees = JSON.stringify(errormsg[0].description, null, 4);
            $('#payment-errors').html( 'Error Code:' + errorcode + ', Error Messages:'
                                     + errorMessagees);
        }
        if (status == 400 || status == 500) {
            $('#payment-errors').html('');
            var errormsg = response.Error.messages;
            var errorMessagees = "";
            for(var i in errormsg){
                var ecode = errormsg[i].code;
                var eMessage = errormsg[i].description;
                errorMessagees = errorMessagees + 'Error Code:' + ecode + ', Error Messages:'
                                     + eMessage;
            }

            $('#payment-errors').html( errorMessagees);
        }
        $form.find('button').prop('disabled', false);
    } else {
        $('#payment-errors').html('');
        var result = response.token.value;
        $('#response_msg').html('Payeezy response - Token value:' + result);
        $('#response_note')
            .html(
                " Note: Use this token for authorize and/or purchase
                transactions. For more details, visit https://developer.payeezy.com/payeezy-api-reference/apis");
        $form.find('button').prop('disabled', false);
    }
};
```

creation is successful, then a status code of “201” is returned. In addition to this, we also display the generated token for your reference using alert() function. Anything other than a “201”, the error message can be

got from the response object. Highlighted code shows how to get the error and the token value(if successful).

The token value will be put in a hidden field (payeezyToken in this case), and then the form will be submitted to the Merchant server for making the payment with the “Token” instead of credit card information.

II. Make a Payment (server side) using the payment token:

Please refer to the Developer portal documentation link <https://developer.payeezy.com/payeezy-api-reference/apis> on how to make a payment ("authorize" or "purchase") with the token.

12. Deploy the Payeezy.js and dependency java script files in the specified folder structure in your website JavaScript deployment folder.

Request payload for Payeezy JS GET – Generate token method

API KEY	Jc9qkh5wkkPxv1K3o807N9wz0qfcHGQE
API SECRET	698189fe9a74ce3627835e594586c8b77e7d1a9aaa129ace9e4123bc13a15747
TA_TOKEN	NOIW
JS_SECURITY_KEY	js-8362201a494a45b6044c1a1aa494e0dcfda71fe5dcab297a
trtoken - Backward Compatible parameter	e37fc002aede8895
Merchant token	fd0a-a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6

Generate Token - Entitlements Removed - v3 - with js_security_key, auth & ta_token	https://api-cat.payeezy.com/v1/securitytokens?apikey=Jc9qkh5wkkPxv1K3o807N9wz0qfcHGQE&js_security_key=js-&callback=Payeezy.callback&type=FDToken&credit_card.type=mastercard&credit_card.cardholder_name=xyz&credit_card.card_number=5424180279791732&credit_card.exp_date=0416&credit_card.cvv=123&auth=false&ta_token=NOIW
--	---

Downloadable information

The sample code for payeezy.js can be found on GitHub. Following is the GitHub link for the sample code.

Click on GitHub link <https://github.com/payeezy/payeezy.js> and look for JavaScript folder to download Payeezy.js file.

Submitting/Generating transaction (method of payments) with example

Payeezy.JS supports the following method of payments

- Credit Card Payments

For API processing details and examples, click here [Method of Payments](#) Or click on the link

<https://developer.payeezy.com/payeezy-api-reference/apis>

Additional capabilities API

- Partner Reporting API - Use our powerful query engine to retrieve payment records. Supports complex filtering, sorting, pagination and more. This is exclusively for Third Party Partners and applicable for a **live** environment only.

For Reporting API processing details, click here <https://developer.payeezy.com/payeezy-api-reference/apis/get/transactions>