

Pay Your Way (PYW) – Implementation Guide

Contents

Introduction	2
Installation	2
1. ReactJS	2
2. Vanilla Java-Script	3
Fields & Attributes Definition	5
API Specifications	7
1. Create/Add Merchant API.....	7
2. Payment Confirmation API.....	8
3. Return Payment API.....	10
Appendix	12

Introduction

Pay Your Way (PYW) provides a payment facility for any online purchase transaction. It provides an instant credit card apply & use facility along with using various other credit cards. PYW also provides payment facility using the *Shop Your Way* (SYW) points.

Installation

PYW is currently available for **ReactJS** & **vanilla Javascript** based web applications.

1. [ReactJS](#)

The `telluride-pyw` React component can be installed in your application using the below `npm` command,

```
npm install --save <file_path>/telluride-pyw-1.1.1.tgz
```

Note:

`<file_path>` (optional), if `telluride-pyw-1.1.1.tgz` file is present in the same location where your application's `package.json` file is kept then you can directly mention the file name and for different location you need to provide full qualified path.

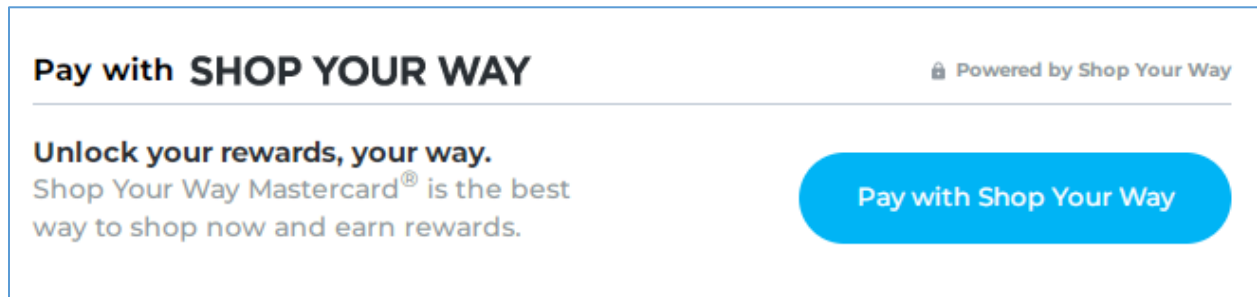
Import `telluride-pyw` component in the appropriate component of your React application.

```
import { Pyw, Refid, Items, Item, updateTotalDue, updateReturnURL } from 'telluride-pyw';
```

For rendering the PYW UI in your application, add the below code snippet at the appropriate level in your code. Please refer [Fields & Attributes Definition](#) section for more details.

```
<div id="pywdiv">
  <Pyw merchant="<client_id attribute received during onboarding process e.g. EXT_RAISE_QA
  >" totalDue="<total due amount to be paid>" returnUrl="<callback url to redirect the control a
  fter payment is complete>" profile="<skip this parameter for production and set value as dev f
  or non-production environment>" >
    <Refid value="<encrypted value of client_id~~access_token~~requestorId~~timestamp~~tr
  ansactionId~~userId>" />
    <Items>
      <Item id="10448" name="Test1" imageUrl="/a/b/c.jpg" qty="2" price="2" unit="usd" />
      <Item id="20448" name="Test2" imageUrl="/a/b/d.jpg" qty="1" price="5" unit="usd" />
    </Items>
  </Pyw>
</div>
```

Once the PYW component is added into your application it looks like below,



2. Vanilla Java-Script

Implement the PYW using plain Java-script by adding the below `<div>` tag code snippet at the appropriate level in the html file.

```
<div id="pyw__widget" class="pyw_widget">
  <input type="hidden" id="refid" value="<encrypted value of client_id~~access_token~~requestId~~timestamp~~transactionId~~userId>" />
  <input type="hidden" id="returnUrl" value="<callback url to redirect the control after payment is complete>" />
  <input type="hidden" id="totalDue" value="<total due amount to be paid>" />
  <input type="hidden" id="merchant" value="<client_id value used in building refid e.g. EXT_RAISE_QA>" />

  <div class="widget">
    <h5>
      <p style="font-weight: 600;">Unlock your rewards, your way</p>
    </h5>
    <div class="widget__header">
      <h4 class="flex">
        <span class="mr-8">Pay with</span>
        <!-- for Production environment, change the below img src value to https://pyweb.telluride.shopyourway.com/pyw_library/images/SYW_logo_black.svg -->
        
      </h4>
    </div>
    <div class="widget__body">
      <div class="description">
        <p class="grey60">The best way to shop now, pay over time, and earn rewards.&nbsp;  <a >Learn more</a>
      </p>
    </div>
    <!-- use "preparePaymentParentWin" function instead of "preparePayment" with the same set of parameters if PYW needs to be opened in the same window -->
    <button type="submit" class="btn btn-primary mt-16 mb-16" onclick="preparePayment(document.getElementById('refid').value, document.getElementById('totalDue').value, document.getElementById('returnUrl').value, document.getElementById('merchant').value)">
      <h4><strong>Pay with Shop Your Way</strong></h4>
    </button>
    <div class="widget__footer">
      <p class="powered">
```

```

        <!-- for Production environment, change the below img src value to https://pywweb.
telluride.shopyourway.com/pyw_library/images/Lock.svg -->
         Powered by Shop Your Way
    </p>
    </div>
</div>
</div>
<div id="pywpopup" class="backdrop">
    <div class="flex center">
        <div class="modal">
            <section class="modal-body">
                <p class="mb-24 flex center">
                    Pay with Shop Your Way in progress...
                </p>
            </section>
        </div>
    </div>
</div>
</div>
</div>

```

Note:

Use below javascript function "preparePaymentParentWin" instead of "preparePayment" if PYW needs to be opened in the same window on click of "Pay with Shop Your Way" button.

```

preparePaymentParentWin(
    document.getElementById('refid').value,
    document.getElementById('totalDue').value,
    document.getElementById('returnUrl').value,
    document.getElementById('merchant').value)

```

There are hidden html elements for capturing refid, returnUrl, totalDue and merchant which are required fields for PYW UI to render using above tags. Please refer [Fields & Attributes Definition](#) section for more details.

- **Link CSS:** please add below css file in your html to render the PYW UI.
(below link is for **UAT** environment)

```

<link href="https://pywweb.uat.telluride.shopyourway.com/pyw_library/styles/pywmain.css" rel="
stylesheet">

```

For production – change CSS href to
https://pywweb.telluride.shopyourway.com/pyw_library/styles/pywmain.css

- **Add script tag:** please add below javascript in your html for the PYW functioning.
(below link is for **UAT** environment)

```
<script src="https://pywweb.uat.telluride.shopyourway.com/pyw_library/scripts/pywscript.js"/>
```

For production – change script src to

https://pywweb.telluride.shopyourway.com/pyw_library/scripts/pywscript.js

Fields & Attributes Definition

Pyw – root element for rendering the PYW UI

merchant – (mandatory) attribute of <Pyw> element. Value of **client_id** attribute assigned to each Client during the onboarding process and the same is used in building the **refid**.

totalDue – (conditional) attribute of <Pyw> element. Total amount to be paid in USD. If

<Items> element is present then **totalDue** can be ignored.

returnUrl – (mandatory) attribute of <Pyw> element. Specify a callback url to redirect the control after payment is complete. It can be a reference path w.r.t. existing client domain (e.g. /checkout/paymentsummary) or a complete http(s) url (e.g. <https://www.client-domain.com/checkout/paymentsummary>).

profile – set this parameter only for non-production environment with value as “dev”, otherwise skip this parameter.

Items – (optional) child element for <Pyw> element. To be added if items need to be shared with PYW to avail any item specific SYW offers or for displaying the item details in the PYW screens.

Item – (mandatory) child element for <Items> element. Capture item details being purchased.

id – (mandatory) attribute of <Item> element. Id of item being purchased.

name – (mandatory) attribute of <Item> element. Name of item being purchased.

imageUrl – (optional) attribute of <Item> element. Logo/image of item being purchased.

qty – (optional) attribute of <Item> element, default “1”. Quantity of item being purchased.

price – (mandatory) attribute of <Item> element. Unit price of item being purchased.

unit – (optional) attribute of <Item> element, default is “usd”. Currency unit for item price.

Refid – (mandatory) child element for <Pyw> element. It is the encrypted value of the following concatenated attributes separated by a delimiter as ~~

```
client_id~~access_token~~requestorId~~timestamp~~transactionId~~userId
```

Where,

- `client_id` – assigned to each Client during the onboarding process.
- `access_token` – computed using a `secret_key` assigned to each Client during the onboarding process (*valid for 2 hours after it has been created*).
- `requestorId` – assigned to each Client during the onboarding process.
- `timestamp` – long value of the current timestamp
- `transactionId` – unique cart id or transaction id or order id maintained by the client
- `userId` – user’s unique id such as user’s login id or email address

Note:

- Use **AES algorithm** to build the encrypted `refid` value
- Refer [Appendix-1](#) to follow the **Client Onboarding process** to get the `client_id` & `secret_key`.
- Refer [Appendix-2](#) to get & use the encryption key required to generate the `refid`.

updateTotalDue – (optional) function to be called on any element or from another reactJS function which changes the total due amount to be paid. This function updates the value of the total due to be paid using `<pyw>` component.

e.g. If there is a text box which is used to change the value of the total due amount then this function **`updateTotalDue(<updated total due amount>)`** can be called on this text box element as below,

```
<input name="totalDue" value={totalDue} onChange={(e) => updateTotalDue(<updated total due amount>)} type="text" />
```

updateReturnURL – (optional) react function to be called on any element or from another reactJS function which changes the Return URL to be redirect once return from pywwweb popup. This function updates the value of the return URL that is used for redirection `<pyw>` component.

e.g. If there is a text box which is used to update the value of the return URL then this function **`updateReturnURL(<updated Return URL>)`** can be called on this text box element as below,

```
<input name="returnURL" value={returnUrl} onChange={(e) => updateReturnURL(<updated Return URL>)} type="text" />
```

API Specifications

1. Create/Add Merchant API

This API can be used to update a few additional details of the existing merchant.

Environment: Available environments are UAT & Production. The details are,

Environment	URL
UAT	https://checkout.uat.telluride.shopyourway.com
Production	https://checkout.telluride.shopyourway.com

Endpoint: `POST /tell/api/merchant/v1/create`

Headers:

- Content-Type: `application/json`
- Accept: `application/json`
- channel: `ONLINE|MOBILE`
- platform: name of the application e.g. `PYW|CHECKOUT`
- merchantClientId: registered client id on Telluride oAuth service
- refid: `client_id~~access_token~~requestorId~~timestamp`
- -- Use **AES algorithm** to build the encrypted refid value using the same secret key received during client onboarding process

Note:

- `requestorId` – can be hardcoded to `PSYW`
- Rest all fields from `refid` are mandatory

Request structure:

Add one or all attributes to be updated

```
{
  "name": merchant name,
  "email": merchant's email address,
  "phone": merchant's phone number(s) - comma separated list if multiple numbers,
  "address": {
    "addressLine1": merchant address line 1,
    "addressLine2": (Optional) merchant address line 2,
    "city": merchant address city,
    "state": merchant address state,
    "postalCode": merchant address postcode,
    "country": (Optional) merchant address country - (default USA)
  },
  "category": merchant category,
  "domains": [ - array: merchant's domain urls to be registered with PYW
```

```

    {"domain": merchant's domain url }
  ],
  "addlAttrs": [ - array: 5 more additional attributes can be added here
    {
      "attr": custom attribute name,
      "val": custom attribute value
    }
  ]
}

```

Response structure & attributes' details:

a) Success Flow:

```

{
  "status": "success"
}

```

b) Error Flow:

```

{
  "status": "error",
  "errors": {
    "code": error code (String)
    "message": error message (String)
  }
}

```

2. Payment Confirmation API

Order payment confirmation API is the handshake API which securely provides the payment details; it can also be used to reverse or cancel the payment.

On successful payment, the PYW returns the control to the client's returnUrl path specified while defining the PYW component by passing a few parameters over the URL such as pywid & pywmsg.

- pywid: unique reference id (UUID) for successful PYW payment. This param is not sent if payment is failed/declined i.e. when pywmsg=N
- pywmsg: Y if the payment is successful otherwise N

Environment: Available environments are UAT & Production. The details are,

Environment	URL
UAT	https://checkout.uat.telluride.shopyourway.com
Production	https://checkout.telluride.shopyourway.com

Endpoint: GET /tell/api/checkout/v1/orderpaymentconfirm

Headers:

- Accept: application/json
- channel: ONLINE
- platform: PYW
- pywid: returned over the returnUrl passed from PYW on successful payment e.g. e.g. 64a12176-8821-4467-9b4d-c8c157a56422
- transactionId: client's transaction id or cart id or order id
- actionType: READONLY (default) / CONFIRM / CANCEL
- transactionType: 1P / 2P - 1st party checkout or 2nd party checkout
- merchantClientId: the same client_id attribute used in building the refid
- refid: refer to **refid** related documentation under "[Fields & Attributes Definition](#)" section below
 - Encrypted value of the combination of client_id~~access_token~~requestorId~~timestamp~~transactionId~~userId
 - Use **AES algorithm** to build the encrypted refid value

Response structure & attributes' details:

c) Response when **actionType=READONLY** (request header)

```
{
  "authCode": unique payment authorization code,
  "orderDate": date of order e.g. 09-30-2021 09:02:10,
  "paymentStatus": payment status e.g. CONFIRMED, CANCELLED
  "paymentTotal": payment amount in USD,
  "paymentDetails": [ --> one or many payments are supported
    {
      "id": payment instance id,
      "type": payment type e.g. SYWR, CREDITCARD,
      "amount": payment amount in USD paid using this payment type,
      "status": payment status for this payment type,
      "currency": payment currency,
      "cardLastFour": credit card last 4 digits - if type=CREDITCARD,
      "cardType": credit card type e.g. Visa - if type=CREDITCARD,
    }
  ],
  "additionalInfo": encrypted value of a string concatenated with
  card#~~exp~~cvv~~zipcode
  -- this can be decrypted by the client using the secret key received
  during the client onboarding process.
  -- it's the same key used to encrypt the refid.
  -- use AES algorithm to decrypt the additionalInfo attribute value
}
```

d) Response when **actionType=CONFIRM** (request header)

```
{
  "authCode": unique payment authorization code,
  "orderDate": date of order e.g. 09-30-2021 09:02:10,
  "paymentStatus": payment status e.g. CONFIRMED, CANCELLED
  "paymentTotal": payment amount in USD,
  "paymentDetails": [ --> one or many payments are supported
```

```

    {
      "id": payment instance id,
      "type": payment type e.g. SYWR, CREDITCARD,
      "amount": payment amount in USD paid using this payment type,
      "status": payment status for this payment type,
      "currency": payment currency,
      "cardLastFour": credit card last 4 digits - if type=CREDITCARD,
      "cardType": credit card type e.g. Visa - if type=CREDITCARD,
    }
  ],
  "additionalInfo": {
    "address": { -> user's address details
      "addressLine1": "17213 Cole Rd",
      "addressLine2": "",
      "state": "MD",
      "city": "Hagerstown",
      "postalCode": "21740",
      "shcAssociateInd": "N",
      "pinNumber": "93536",
      "zipCodeExtension": ""
    }
  }
}

```

3. [Return Payment API](#)

This API can be used to return the full or partial payment for the given payment authcode.

Environment: Available environments are UAT & Production. The details are,

Environment	URL
UAT	https://payment.uat.telluride.shopyourway.com
Production	https://payment.telluride.shopyourway.com

Endpoint: **POST** /tell/api/payment/v1/return

Headers:

- Content-Type: application/json
- Accept: application/json
- channel: ONLINE
- platform: PYW
- merchantClientId: registered client id on oAuth service
- transactionId: Client transaction ID e.g. 10001
- refid: client_id~~access_token~~requestorId~~timestamp~~transactionId~~userId
 -- Use **AES algorithm** to build the encrypted refid value using the same secret key received during client onboarding process

Note:

- requestorId – PSYW
- userId – can be empty or ignored
- Rest all fields from refid are mandatory

Request structure:

```
{
  "authCode": from orderpaymentconfirm API response e.g. 09801-20211117-1020292,
  "returnAmount": amount to be returned for already made payment
    - if returnPayments attribute (below) details are not provided then
    return amount is adjusted on prorata basis against the tenders used for
    making the payment,
  "returnPayments": (optional) [ - array: one or more payments to be returned
    {
      "id": payment id from orderpaymentconfirm API response,
      "amount": payment amount to be returned for the given payment id
    }
  ]
}
```

Response structure & attributes' details:

a) Success Response

```
{
  "returnCode": "90128789-34567892",
  "returnDate": datetime on which the payment is returned
  "returnPaymentStatus" : [ - array: one or many returned payment instances here
    {
      "id": payment id which is returned,
      "type": payment type e.g. SYWR|CREDITCARD,
      "amount": payment amount in USD paid using this payment type,
      "currency": payment currency,
      "cardLastFour": credit card last 4 digits - Only if type=CREDITCARD,
      "cardType": credit card type e.g. Visa - Only if type=CREDITCARD
    }
  ],
  "status": "success"
}
```

b) Error Response

```
"status": "error"
"errors": {
  "code": error code (String)
  "message": error message (String)
}
```

Appendix

1. Client onboarding document (Merchant_Onboarding_v1.5.pdf - refer from enclosing zip file)
2. Document for building & encrypting the refid (Refid_Encryption_Key_v1.0.pdf - refer from enclosing zip file)
3. To add a new Public Key / Certificate renewal
 - a. Login to OAuth Portal.
 - If you don't have Password please use the "forgot password" option to reset the password using your client_id/merchant_id & registered email.
 - OAuth Portal URL,
 - UAT: <https://oauthweb.uat.telluride.transformco.com/oauthUI/login>
 - Prod : - <https://oauthweb.telluride.transformco.com/oauthUI/login>
 - b. After Login you will see (+) sign next your existing certificate rows. Click the (+) sign to add new Public Key.