



PaywayWS Redirected Payment Form Integration Guide

Date: September 11, 2020

NOTICE:

This technical specification is the confidential and proprietary product of Payway®, Inc. Any unauthorized use, reproduction, or transfer of this specification is strictly prohibited. All rights reserved.

Copyright © 2021 by Payway®, Inc
Apple Pay® is a registered trademark of Apple Inc

Table of Contents

REVISION HISTORY	3
RELATED DOCUMENTS	3
INTRODUCTION	3
I. BROWSER ACCESSIBLE DATA	4
II. INTEGRATION STEPS FOR PROCESSING A REDIRECTED ACCOUNT ENTRY	5
A. QUEUE TRANSACTION	7
B. SEND QUEUED TRANSACTION	7
C. REDIRECTED ACCOUNT RESULTS	8
III. MANAGING PAYWAY TRANSACTIONS	8
IV. APPENDIX A HOSTEDPAY.JS	9

Revision History

Revision	Date	Initials	Comments
524	7/1/2019	DN	Initial Revision.
528	9/9/2020	DRF	Added Java script to Appendix

Related Documents

- PaywayWS ACH Integration Guide
- PaywayWS Credit Card Integration Guide
- Payway® Users Guide
- Payway® Admin Guide
- Payway® ApplePay Integration Guide
- Payway® GooglePay Integration Guide
- Payway® Point-to-Point Encryption (P2PE) User's Guide
- Payway® File Processing Users Guide

Introduction

This document provides a guide to integrating Payway's Redirected Payment Form. This feature allows a client to securely redirect credit card and bank account data input to the PaywayWS host, eliminating the need to manage this data on your server. PaywayWS provides the restful web service with which you will interact to implement Hosted Payment Entry. A Payway supplied java script handles the redirection of a client's browser to PaywayWS for account data input.

Note: See the section titled “interactive requests” in the PaywayWS Credit Card Integration-Guide.pdf for details on the JSON requests that support hosted payment transactions. This guide gives a general overview of the transaction flow and architecture for implementation.

I. Browser Accessible Data

Security is achieved within the Hosted Payment Entry environment by limiting the information that is sent to the browser where users can intercept and manipulate the information. During the checkout, account entry, and payment processes the browser has access to only three pieces of data in the course of processing a transaction. These are described in the table below.

Name	Description
paywayRequestToken	This token can be used only in relation to the current transaction. This means it can be used only once, and the transaction data cannot be changed since it was set up during the prior queue transaction request. See the sample application and PaywayWS Credit Card Integration guide for queue transaction examples.
paywayCode	This is the status of the completed transaction.
paywayMessage	Any message associated with the completed transaction, either empty or a further explanation of the paywayCode.

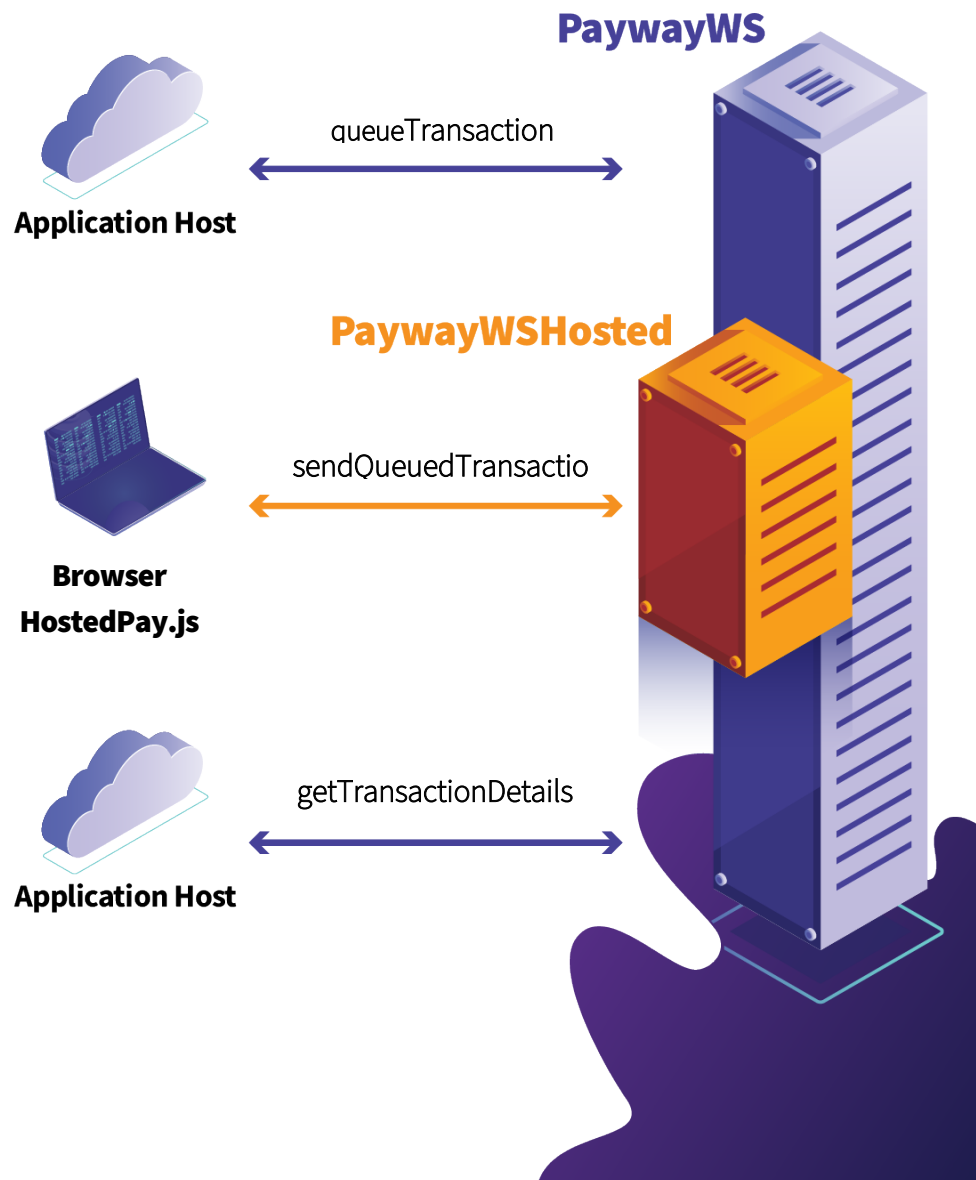
II. Integration Steps for Processing a Redirected Account Entry

Processing a Redirected Account Entry transaction is done in three steps. The first step and the final step are executed from your server and are completed via PaywayWS requests. The middle step is handled by the Redirected Account Entry java script (HostedPay.js). The steps are described in the sections below. During a Redirected Account Entry transaction only the transaction token, transaction result code, and possible error message is available to the browser.

The block diagram below shows the steps for a Redirected Account Entry transaction. Note that the middle step is done by the java script.

WARNING: DO NOT EXECUTE STEPS 1 OR 3 FROM A JAVA SCRIPT. THESE INVOLVE DATA THAT SHOULD NOT BE PASSING THROUGH THE CLIENT BROWSER.

Redirected Account Entry Transaction Steps



A. Queue Transaction

Note: See PaywayWS Credit Card Integration Guide.pdf in the section titled “Interactive Requests” for details on the queue transaction request.

Before loading a payment page to enable the user to enter card account information, you will send PaywayWS a request to queue a payment. This request returns a `paywayTransactionToken` that is sent to the browser and used to complete the payment. The transaction token is like a PaywayWS session token, except that it can be used only to interact with the queued transaction. This allows Payway to send to the browser the transaction token without concern for a Payway session being hijacked.

The queued payment includes the payment amount, source id, and payment type. Once this step is completed, your Payment Page can be presented. See the PaywayWS Credit Card Integration Guide and the sample application for more on sending the queue payment request.

B. Send Queued Transaction

Note: See PaywayWS Credit Card Integration Guide.pdf in the section titled “Interactive Requests” for details on sending a queued transaction. At the beginning of implementation, Payway will provide you with a java script called `HostedPay.js`. This will contain sample javascript code to send the queued transaction.

When the user authorizes the payment on the payment page you present, the java script will send the account information and `paywayRequestToken` to PaywayWS for processing. The results are processed by the java script. Only the result code and any related error message are sent back to the browser. Detailed transaction results can be requested from your server after the transaction is completed (the java script posts the result code to your server).

Note that the transaction token can be used only to send the queued transaction and read the results.

C. Redirected Account Results

Note: See PaywayWS Credit Card Integration Guide.pdf in the section titled “Interactive Requests” for details on the hostedPaymentResults request that returns details for a hosted payment.

In response to the request to send the queued transaction (sendQueuedTransactionPaywayWS request), PaywayWS returns the result code and any error message. This may be all you need to complete the transaction. The java script manages the sendQueuedTransaction request and posts the result code and paywayMessage back to your server. Optionally you can query the transaction details by sending a hostedTransactionResults request to PaywayWS.

III. Managing Payway Transactions

To manage your transactions, you can use PaywayWS requests. These include reading transactions and updating user accounts where needed.

Reports and other payment management functionality is provided by the Payway web application. See the Payway User’s Guide for more information.

IV. Appendix A HostedPay.js

```
// direct the payment request with PAN data to the target PaywayWS server
function HostedPaymentClicked ()
{
    console.log ( "HostedPaymentClicked" );

    // java script http request object.
    var request = new XMLHttpRequest();

    // values for the request come from the jsp page.
    try
    {
        // Set up the json request
        var requestjson =
        {
            accountInputMode : "primaryAccountNumber",
            paywayRequestToken :
document.getElementById('paywayRequestToken').value,
            transactionName : document.getElementById('transactionName').value,
            transactionSourceId : document.getElementById('transactionSourceId').value,
            request : "sendQueuedTransaction",

            cardAccount :
            {
                accountNumber : document.getElementById ( 'accountNumber' ).value,
                expirationDate : document.getElementById ( 'expirationDate' ).value,
                firstName : document.getElementById('first').value,
                lastName : document.getElementById('last').value,
                address : document.getElementById('address').value,
                city : document.getElementById('city').value,
                state : document.getElementById('state').value,
                zip : document.getElementById('zip').value,
```

```
        email : document.getElementById('email').value,
        phone : document.getElementById('phone').value,
        accountNotes1: "notes1",
        accountNotes2: "notes2",
        accountNotes3: "notes3"
    }
};

console.log ( JSON.stringify ( requestjson ) );

console.log ( document.getElementById('serviceLocation' ).value );

// send the request to PaywayWS
request.open('POST', document.getElementById('serviceLocation').value +
'/PaywayWS/Payment/CreditCard', true );
request.setRequestHeader("Content-type", "application/json");
request.send( JSON.stringify ( requestjson ) );
}
catch ( err )
{
    alert ( err.message );
    console.log ( err.message );
}

request.onload = function()
{

    try
    {
        // Check the results of the request using request.responseText. Here
        // we just forward it to a results page which prints out the returned JSON object.

        // debugging
        console.log ( request.responseText );

        // set the response in the current document and submit to sample confirmation
```

```
page
    document.getElementById('results').value = request.responseText ;
    document.getElementById("ResultsTransaction").submit() ;
}
catch ( err )
{
    // error processing, here just log to the console
    console.log ( "Submit page error: " + err ) ;
}

};

request.onerror = function()
{
    console.log ( "Payment error: " + request.responseText ) ;
};

}
```