



Apple Pay[®] Integration Guide

Date: July 1, 2019
Version 524-1.1

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.

Table of Contents

REVISION HISTORY	3
INTRODUCTION	4
I. APPLE PAY AND 3DSECURE	4
II. INTEGRATION METHODS	4
A. USING PAYWAY APPLE PAY JAVASCRIPT	4
B. PAYWAYWS APPLE PAY CRYPTOGRAM REQUESTS	4
III. INTEGRATION STEPS FOR PROCESSING PAYWAY APPLE PAY	5
A. QUEUE TRANSACTION	6
B. VALIDATE APPLE PAY MERCHANT	6
C. AUTHORIZE APPLE PAY PAYMENT	6
D. QUERY TRANSACTION DETAILS	6
IV. MANAGING PAYWAY TRANSACTIONS	7
V. SAMPLE APPLICATION	7
VI. APPLE®, INC. REQUIREMENTS FOR APPLE PAY	8

Revision History

Revision	Date	Initials	Comments
524	7/1/2019	DN	Initial Revision.

Related Documentation

- PaywayWS Integration Guide
- Payway® Administrator's Guide

Introduction

This document provides a guide to integrating Apple Pay into your web applications. Apple Pay provides a secure, wallet-based payment method for your users. PaywayWS provides the RESTful web service with which you will interact to implement your Apple Pay interface. The actual integration is done in a JavaScript that we provide to you and you can customize to your needs.

I. Apple Pay and 3DSecure

To use Apple Pay, a user must register a credit card in the wallet on their Apple device. When this process is complete, the card is registered on a 3DSecure token server and the related account number and OnlinePaymentCryptogram is stored securely on the phone. Note that the account number on the wallet entry is not the same as the credit card account number. The data from the wallet on the Apple device is what Payway® sends to the card processor. The processor then interacts with the token server to acquire the original credit card PAN, and the transaction is processed.

II. Integration Methods

There are two ways to integrate Apple Pay transactions with Payway.

A. Using Payway Apple Pay JavaScript

Our JavaScript contains all the logic necessary to communicate with PaywayWS, decrypt Apple Pay payloads, and process payments. We take care of all the cryptography and certifications with Apple Pay. We provide you with a standard JavaScript which you can then modify to support your own payment sheet that is displayed to the user.

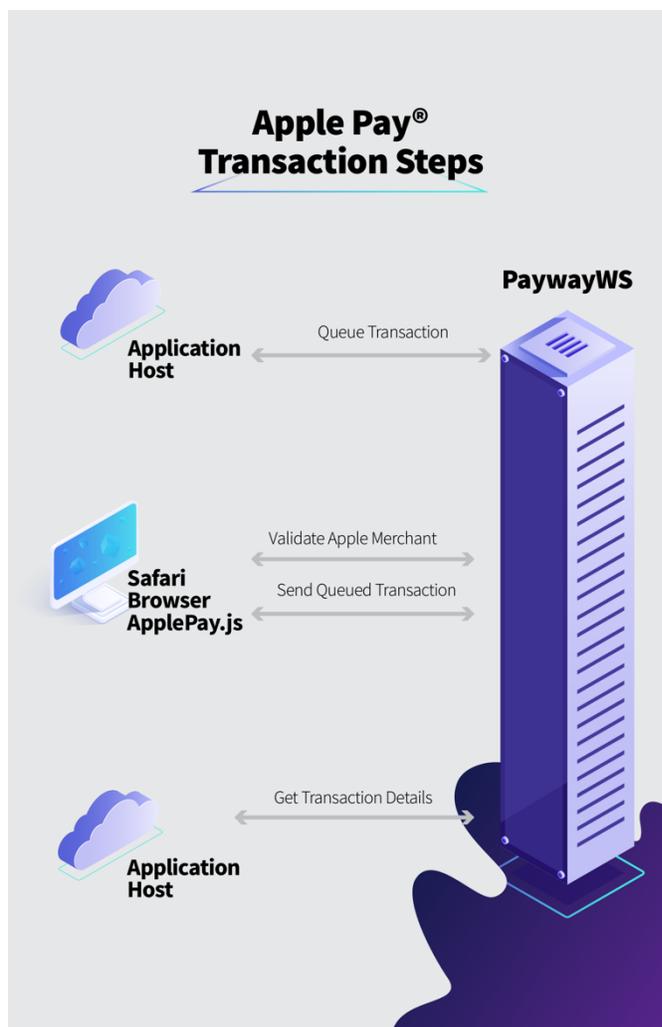
B. PaywayWS Apple Pay Cryptogram Requests

In some cases, our partners choose to implement their own Apple Pay integration. This requires managing of the Apple Pay merchants, processors, certificates, etc. In this mode you will send decrypted Apple Pay payloads directly to PaywayWS for processing. See the PaywayWS document for information on the ApplePayCryptogram account entry type. The remainder of this document describes the steps required to integrate with the Apple Pay JavaScript discussed in the previous section.

III. Integration Steps for Processing Payway Apple Pay

Processing an Apple Pay payment is done in four steps. The first step and the final step are executed from your server and are completed via PaywayWS requests. Step 2 and 3 are handled in the Apple Pay JavaScript. The steps are described in the sections below. Note that the first and last steps are executing from your server to the Payway® server. These steps include data that is not sent to the browser. During an Apple Pay transaction only the transaction token, transaction result code, and any error message is available to the browser. The transaction token is described later in this section.

Below is a block diagram of the steps for an Apple Pay transaction. Note the two middle steps on the left side are initiated by the JavaScript that we provide for integration.



DO NOT EXECUTE STEPS 1 OR 4 FROM A JAVASCRIPT. THESE INVOLVE DATA THAT SHOULD NOT BE PASSING THROUGH THE CLIENT BROWSER.

A. Step One: Queue Transaction

Before presenting the page with the Apple Pay button, you will send PaywayWS a request to queue a payment. This request will return a `paywayTransactionToken` that will be 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 us to send to the browser this transaction token without concern for hijacking a Payway session. The queued payment includes the payment amount, source id, and payment type. Once this step is completed the Apple Pay button can be displayed on your page. See the PaywayWS integration guide and the sample application for more on sending the queue payment request.

B. Step Two: Validate Apple Pay Merchant

When the user presses the Apple Pay button on your page, the Apple Pay payment sheet is displayed and a request is sent to PaywayWS to validate Payway® as a valid Apple Pay provider. The payment sheet is ready to accept payment authorization from the user.

C. Step Three: Authorize Apple Pay Payment

Depending on your customized modifications to the JavaScript, the user may interact with the payment sheet by modifying certain fields. When the user authorizes the payment, the JavaScript will send the encrypted Apple Pay information to PaywayWS for processing. The results are processed by the JavaScript. In the sample code the JavaScript posts the results to a confirmation page. You will need to interpret the results and take appropriate action. Note that the JavaScript sends the transaction token to refer to the payment that was queued in step 1. Only the transaction token is available to the browser and the transaction token can be used only to send the queued transaction and read the results.

D. Step Four: Query Transaction Details

In response to the request to send the queued transaction (`sendQueuedTransaction` PaywayWS request), PaywayWS returns the result code and any error message. This may be all you need to complete the transaction. Optionally you can query the transaction details by sending a request to PaywayWS, `getPaymentDetails`. An example of this is in the sample application, the `TransactionDetails.jsp` page. This will return all fields for the payment and account records, with PAN data masked.

At this point you can store and/or present any of this information to the user for receipt purposes or other processing. This interaction is between your server and PaywayWS so none of this information is available to the browser during the transaction.

IV. Managing Payway Transactions

The Payway Apple Pay integration uses the PaywayWS RESTful API to validate Apple Pay merchants and perform Apple Pay 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.

V. Sample Application

There is a sample application in the form of .jsp pages and a JavaScript. The following files constitute the sample application. The files are commented on to explain each step and indicate where you can customize the JavaScript for your needs.

File	Description
Cart.jsp	Simple shopping cart page containing amount and transaction type specification (authorize, sale, credit).
Checkout.jsp	Payment page including Apple Pay button. This is where the queue transaction requests happen, sending the transaction information to PaywayWS. This page includes calls to the JavaScript to execute the payment
TransactionResults.jsp	This page simply presents the result of the payment, the PaywayWS code and any error message that was returned to the browser.
TransactionDetails.jsp	Optional page that requests all payment details. Note this is executed from the host application server to PaywayWS, NOT from the client browser.
PaywayApple.js	Where the Apple Pay merchant validation and transaction are completed. This is the JavaScript you can modify to suit your application needs.
PaywayApple.css	Styles for the Apple Pay button.

VI. Apple®, Inc. Requirements for Apple Pay

Apple has requirements for the look and feel of the button, placement of the button, and interaction with the payment sheet. Refer to Apple's web site for integration guides including Apple requirements on presenting the Apple Pay button.

<https://developer.apple.com/design/human-interface-guidelines/apple-pay/overview/buttons-and-marks/>