inovio

Merchant onboarding for Apple Pay

Client Setup User Guide

Version 2.0

Table of Contents

1 Overview

1.1 Client Documentation

This documentation is intended for merchants who will process transactions with Apple Pay and should be used as a guide on Apple Pay configuration through the Portal and the client server side setup.

1.2 Revision History

Version	Date	Description
1.0	23 Aug 2023	Initial version
1.1	30 Aug 2023	Added Apple Pay and Inovio's IP Addresses to be whitelisted.
1.2	07 Sept 2023	Added Payment Page Implementation Section
1.3	19 Sept 2023	Enhanced Syntax Color Scheme
1.4	02 Oct 2023	Corrected Apple Pay "requirements" URL, Added Domain Association Note
1.5	12 Nov 2023	Added a note to indicate that real PAN is required for Production Testing
1.6	10 May 2024	Corrected typo errors on Endpoint URL from: from https://api.inoviopay/payment/applepay.cfm to https://api.inoviopay.com/payment/applepay.cfm and FetchUrl from https://[domain]/apple-pay-poc/api/session/create to https://[[DOMAIN]]/apple-pay-services/api/session/create
2.0	19 Dec 2024	Added 3rd party token support

2 Apple Pay Domain Registration Process

2.1 Requirements

This implementation is for web only integration setup without an In-App Purchase option

Merchants will use Inovio API and use either Inovio's Hosted checkout payment page or their own payment form through their checkout page. App based support will not be part of this integration.

- Apple Pay configuration shall be available to all clients
- Merchants will need to add the Apple Pay library to their website to make calls to the Apple Pay API) – *Instructions to be provided through a separate Technical Documentation.*
- Merchants will be using Inovio's direct API integration with Apple Pay

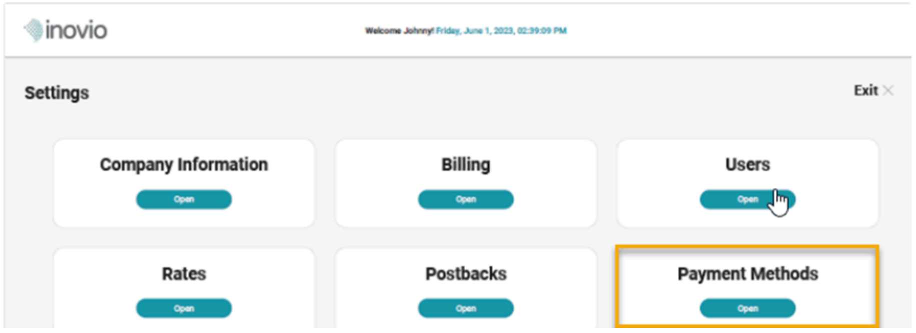
- Merchants will not need to create a Developer AccountMerchant ID, a Merchant ID certificate or a Payment Processing Certificate with Apple Pay
- Your domain must have a valid SSL certificate.
- All pages that include Apple Pay must be served over HTTPS.
- Merchants located in the following countries can accept Apple Pay.
- Customers whose issuing banks are located on this list are eligible to purchase with Apple Pay.

The following instructions are meant as a guide to clients on the necessary steps to setup Apple Pay through the Portal by registering the domains on which the client will display Apple Pay button. Clients will need to register all their domains that will be displaying the Apple Pay button, including both top-level domains (e.g. http://example.com) and sub-domains (e.g. store.example.com).

- This is the initial step before additional work is done on the client's server.
- Clients are to Whitelist/Allow the following Apple IP address ranges for Domain Verifications if their domain is protected from public access
- Clients shall Whitelist/Allow the following Inovio's IP addresses
  - 69.165.111.144
  - 66.23.212.68
  - 66.23.212.69
  - 155.130.131.68
  - 155.130.131.69
  - 66.116.108.128/26 (range)
  - 66.23.214.192/26 (range)
- This process is only necessary when a client is using their own checkout/payment page.

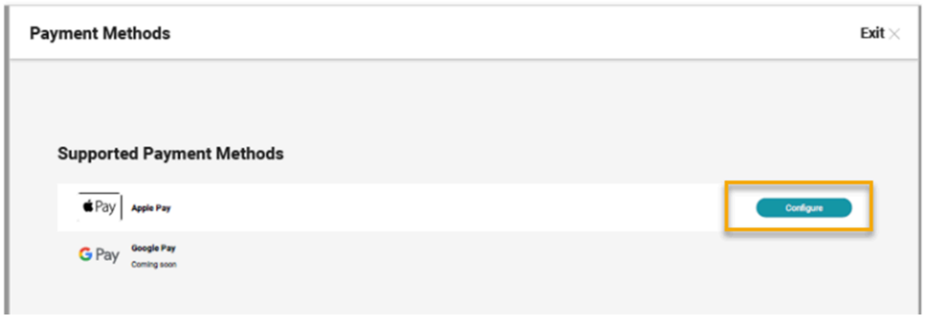
2.1.1 Open Payment Methods Page

- In order to access Supported Payment Methods where Clients will be able to launch Apple Pay Configuration, clients will need to open Payment Methods Tile located within the Settings Menu
- Opening this tile opens the Payment Methods Page
- Currently enabled Payment method within this tile is Apple Pay only.



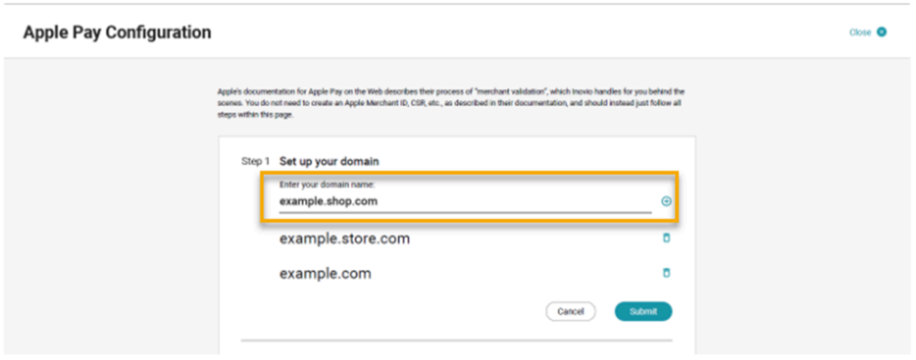
2.1.2 Launch Apple Pay Configuration Page

- Apple Pay Configuration is for clients to add the domains where Buy with Apple Pay will be displayed
- Clicking "**Configure**" will open the Setup your domain window.



2.1.3 Add Domains and Sub-domains.

- Here, Clients will click Add, delete change domains associated with Apple Pay
- The +/Add sign allows clients to type their domains in the provided interface
- Clients will be able to add more than one domain with the +/Add button.
- Clicking “Cancel” shall exit the Setup window and back to Configure screen (First screen)
- “Submit” button shall be greyed out until there’s a domain entered and ready to submit
- Clients should not include “HTTPS” when adding a domain

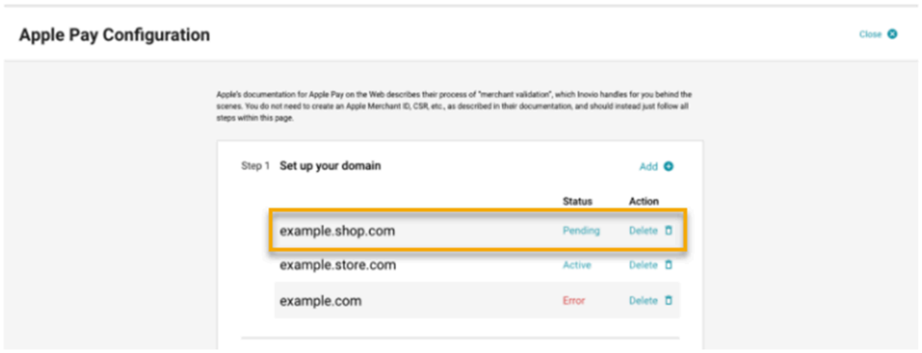


Apple's documentation for Apple Pay on the Web describes their process of "merchant validation", which Inovio handles for you behind the scenes. You do not need to create an Apple Merchant ID, CSR, etc., as described in their documentation, and should instead just follow all steps within this page.

2.1.3.1 Domain Pending Status

Once a domain has been submitted, it’s immediately put on Pending Status. The domain will be on a pending status until the domain association file has been uploaded and the domains associated through */well-known/apple-developer-merchantid-domain-association*

- Once a domain has been added, domain will show “Pending” Status (*This is waiting for Domain Association File to be uploaded and hosted*)
- Note! The domain file will be checked from both Inovio’s servers and Apple servers. Your file MUST allow access from this **list of IP addresses**
- A list of submitted domain(s) will be shown along with the status of each domain.



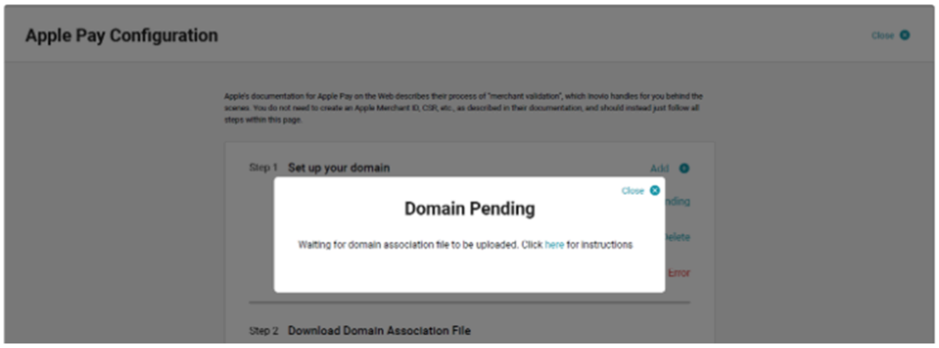
Apple Pay Configuration

Apple's documentation for Apple Pay on the Web describes their process of "merchant validation", which Inovio handles for you behind the scenes. You do not need to create an Apple Merchant ID, CSR, etc., as described in their documentation, and should instead just follow all steps within this page.

Step 1 Set up your domain

	Status	Action
example.shop.com	Pending	Delete
example.store.com	Active	Delete
example.com	Error	Delete

- Clicking the "Pending" link opens up a pop-up message informing the Client that they have a step remaining to complete the Apple Pay Setup



Apple Pay Configuration

Apple's documentation for Apple Pay on the Web describes their process of "merchant validation", which Inovio handles for you behind the scenes. You do not need to create an Apple Merchant ID, CSR, etc., as described in their documentation, and should instead just follow all steps within this page.

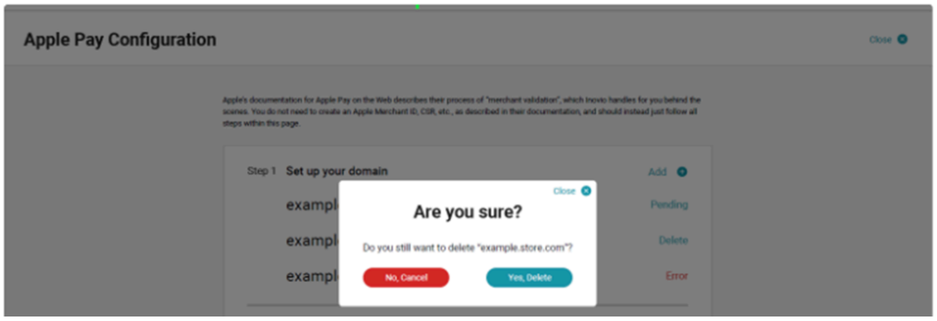
Step 1 Set up your domain

Domain Pending

Waiting for domain association file to be uploaded. Click here for instructions

Step 2 Download Domain Association File

- If no new domain is being added, clients can click the hyperlinked text to view Domain Association File upload instructions again
- Clicking the "Delete" a domain shall lead to confirmation prompt as below before the domain is deleted



Apple Pay Configuration

Apple's documentation for Apple Pay on the Web describes their process of "merchant validation", which Inovio handles for you behind the scenes. You do not need to create an Apple Merchant ID, CSR, etc., as described in their documentation, and should instead just follow all steps within this page.

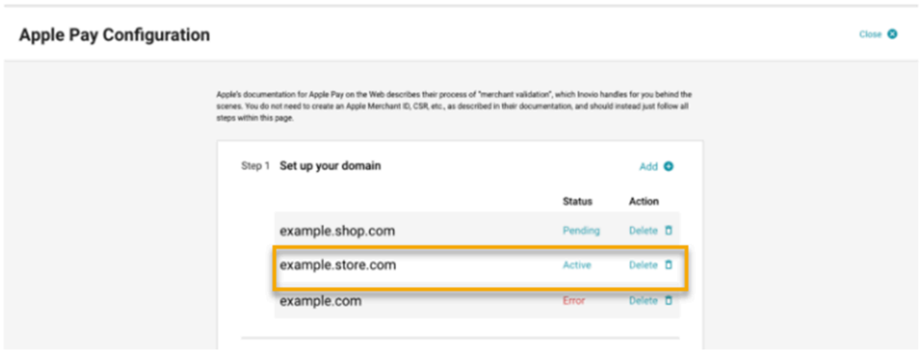
Step 1 Set up your domain

Are you sure?

Do you still want to delete "example.store.com"?

2.1.3.2 Domain Active Status

Once a domain has been verified, the status changes to Active. This means that the domain is ready for additional configuration backend work.


Apple Pay Configuration

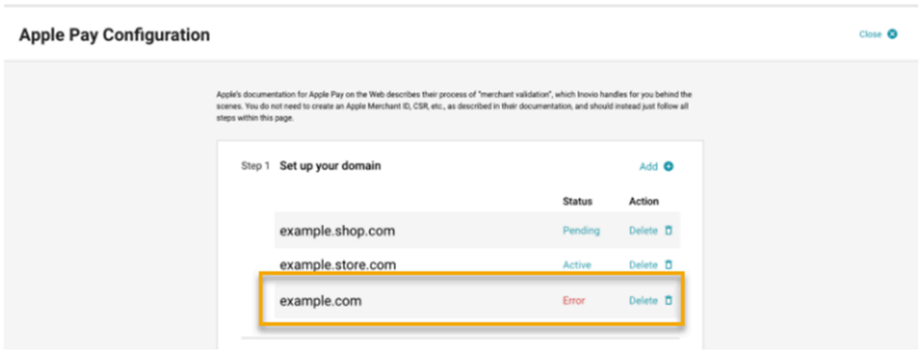
Apple's documentation for Apple Pay on the Web describes their process of "merchant validation", which Inovio handles for you behind the scenes. You do not need to create an Apple Merchant ID, CSR, etc., as described in their documentation, and should instead just follow all steps within this page.

Step 1 Set up your domain

	Status	Action
example.shop.com	Pending	Delete
example.store.com	Active	Delete
example.com	Error	Delete

2.1.3.3 *Domain Error Status*

In the event there are errors during verification process, the domain status will be set to error and the respective error will be displayed through the integrated link.

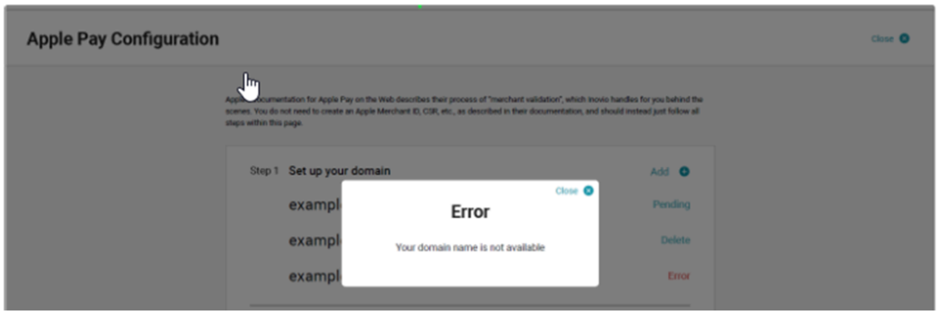

Apple Pay Configuration

Apple's documentation for Apple Pay on the Web describes their process of "merchant validation", which Inovio handles for you behind the scenes. You do not need to create an Apple Merchant ID, CSR, etc., as described in their documentation, and should instead just follow all steps within this page.

Step 1 Set up your domain

	Status	Action
example.shop.com	Pending	Delete
example.store.com	Active	Delete
example.com	Error	Delete

- Clicking the error link shall open a pop-up window with additional information on the related error.


Apple Pay Configuration

Apple's documentation for Apple Pay on the Web describes their process of "merchant validation", which Inovio handles for you behind the scenes. You do not need to create an Apple Merchant ID, CSR, etc., as described in their documentation, and should instead just follow all steps within this page.

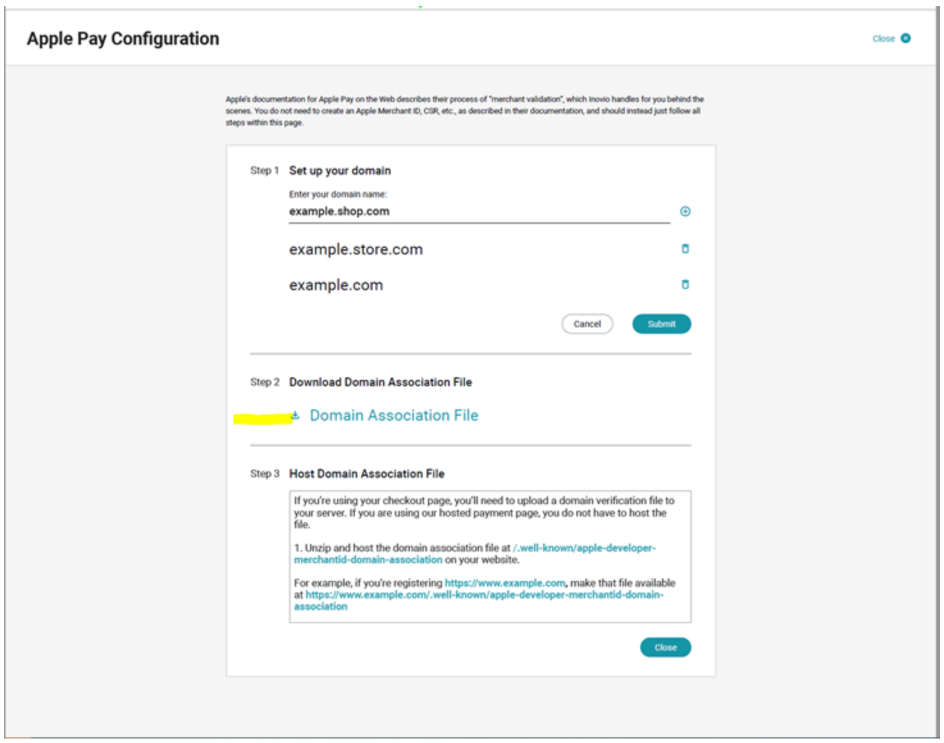
Step 1 Set up your domain

Error

Your domain name is not available

2.1.4 **Download Domain Association File**

- Domain Association File shall be available for client download through the "Download Association File" link.
- This file remains consistent and does not change.
- The file being downloaded shall be in a text format
- Clients will need to download the Domain Association File and host it as instructed



Apple Pay Configuration

Apple's documentation for Apple Pay on the Web describes their process of "merchant validation", which Inovio handles for you behind the scenes. You do not need to create an Apple Merchant ID, CSR, etc., as described in their documentation, and should instead just follow all steps within this page.

Step 1 Set up your domain

Enter your domain name: example.shop.com

example.store.com

example.com

Step 2 Download Domain Association File

Step 3 Host Domain Association File

If you're using your checkout page, you'll need to upload a domain verification file to your server. If you are using our hosted payment page, you do not have to host the file.

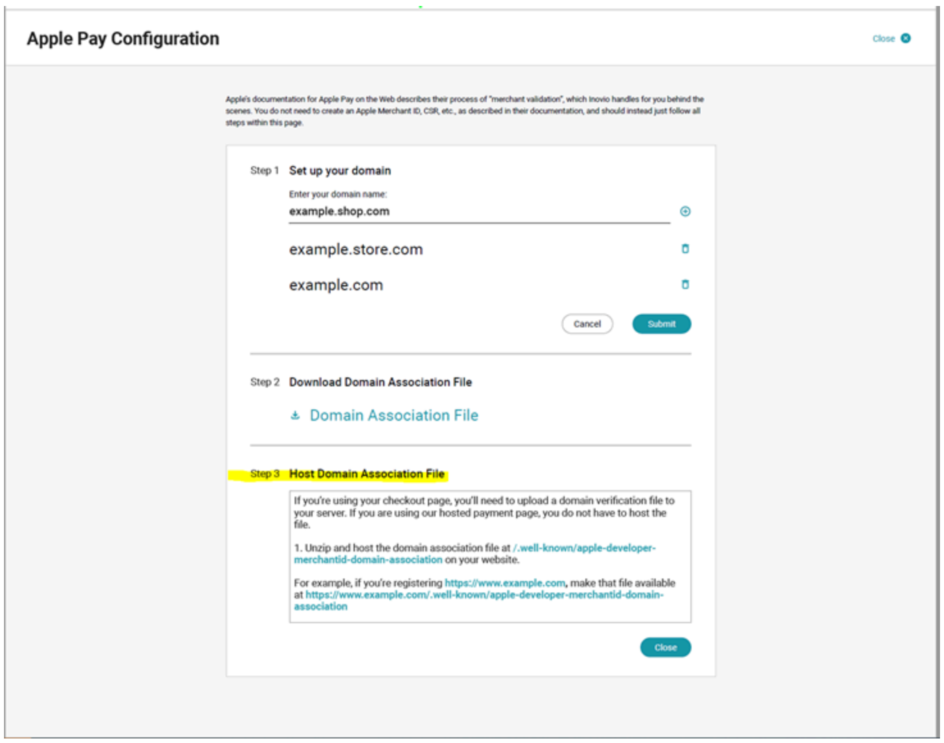
1. Unzip and host the domain association file at /.well-known/apple-developer-merchantid-domain-association on your website.

For example, if you're registering https://www.example.com, make that file available at https://www.example.com/.well-known/apple-developer-merchantid-domain-association

2.1.5 Domain Association File Instructions

- Instructions on how to host the domain association file are part of this UI.
- Clients shall be instructed on how to upload the domain association file to *\/well-known/apple-developer-merchantid-domain-association*

NOTE: Domain Association File does not change and should be stored in the server as long as Apple Pay service is enabled.


Apple Pay Configuration

Apple's documentation for Apple Pay on the Web describes their process of "merchant validation", which Inovio handles for you behind the scenes. You do not need to create an Apple Merchant ID, CSR, etc., as described in their documentation, and should instead just follow all steps within this page.

Step 1 Set up your domain

Enter your domain name: example.shop.com

example.store.com

example.com

Step 2 Download Domain Association File

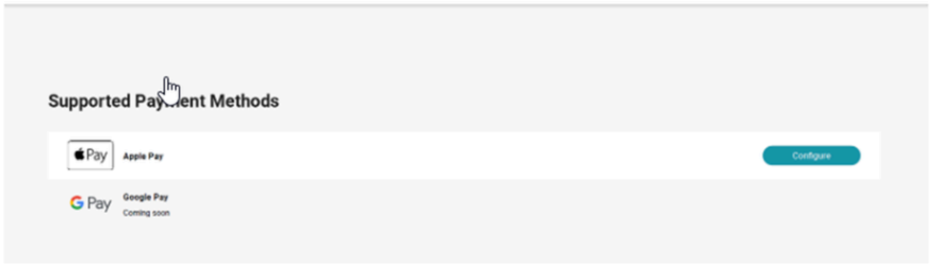
Step 3 Host Domain Association File

If you're using your checkout page, you'll need to upload a domain verification file to your server. If you are using our hosted payment page, you do not have to host the file.

1. Unzip and host the domain association file at /.well-known/apple-developer-merchantid-domain-association on your website.

For example, if you're registering https://www.example.com, make that file available at https://www.example.com/.well-known/apple-developer-merchantid-domain-association

- Close button shall close the setup window and sent the customer back to the "Configuration" Window



3 *Apple Pay Payment Page Setup*

3.1 Overview

This section discusses how to send credit card transactions to the Payment Gateway using Apple Pay. In order to do so, the web pages which host the payment forms must have their domains registered. See Section 2 for more details

Make sure your website fulfills Apple Pay's requirements before getting started.

This section assumes that the domain registration process and host the verification file download has been completed.

Additionally, the Processor and Merchant Account being used must also support Apple Pay. Contact your support team for more information.

3.2 Implementation Process Flow

This is a list of the order in which you can set up the things needed to process transactions with Apple Pay.

- Fetch the Apple Pay configuration data from the gateway
- Update the configuration with the specific payment/order details
- Present the Apple Pay button
- Add the Apple Pay library to the payment page
- Create a JS function to validate your domain
- Create and start the Apple Pay session
- Get the authorized payment information from the customer's browser
- Send the transaction to the gateway

Each item in the list is discussed in the following sections, and some example code is provided. Note that all code provided is for use as an example only, and does not represent code that can be used in your production environment.

NOTE: Real PAN must be used when testing in production environment. Test cards will not work.

3.3 Fetch Apple Pay configuration data from the gateway

You will need to get configuration data from the gateway in order to set up the Apple Pay session when the customer clicks the Apple Pay button. This comes from a request to the following endpoint:

Endpoint: https://api.inoviopay.com/payment/applepay.cfm

- **Requests must be made in JSON format.**

Parameters:

Field Name	Description
REQ_USERNAME	API CREDENTIAL USERNAME
REQ_PASSWORD	API CREDENTIAL PASSWORD
DOMAIN_NAME	DOMAIN NAME WHERE THE PAYMENT PAGE IS HOSTED/SERVED FROM
CLIENT_ID	CLIENT ID
REQUEST_ACTION	REQUEST_ACTION = APPLEPAYCONFIG USED TO INSTRUCT THE ENDPOINT TO PROVIDE THE APPLE PAY CONFIGURATION

Note: All parameters are required

The response will be a JSON-formatted string. Here is an example of the structure:

```
{
  "APPLEPAY_CLI_CONF_ID": xx, // internal unique id; do not change
  "REQ_ID": 11111111111, // trace ID, used for troubleshooting
  "CLIENT_ID": 1111111, // your gateway api client_id
  "INITIATIVE": "web", // identifies e-commerce transaction
  "DISPLAY_NAME": "xxxxxxx", // Short, localized description of the merchant.
  "PARTNER_MERCH_NAME": "xxxxxx", //name of the merchant
  "PARTNER_INTERNAL_MERCH_ID": "xxxxxxxxxxxxx", // identifies the merchant to
apple
  "ENCRYPT_TO": "xxxxxxxxxxxxxxxxxxxx", // merchant ID to Apple
  "DOMAIN_LIST": [ // domain registered in inovio portal serving this page
    {
      "DOMAIN_NAME": "paymentpagedomain.com",
      "DOMAIN_STATUS": "ACTIVE"
    }
  ],
  "paymentRequest": { //
    "countryCode": "US",
    "currencyCode": "USD",
    "merchantCapabilities": [
      "supports3DS"
    ],
    "supportedNetworks": [
      "visa",
      "masterCard"
    ],
    "requiredBillingContactFields": [
      "postalAddress",
      "name",
      "phoneticName",
      "phone",
      "email"
    "total": {
      "label": "" // required; a short, localized description of the line item
      "type": "final", // do not change
      "amount": 0 // must be greater than or equal to zero        }
    },
  },
  "fetchUrl": "https://[[DOMAIN]]/apple-pay-services/api/session/create"
}
```

This should be provided on your payment page in the parameter: merchantConfig

Example:

```
<script>
  this.merchantConfig = {
    "APPLEPAY_CLI_CONF_ID": 11,
    "REQ_ID": 11111111111,
    "CLIENT_ID": 1111111,
    "INITIATIVE": "web",
    "DISPLAY_NAME": "xxxxxxx",
    "PARTNER_MERCH_NAME": "xxxxxx",
    "PARTNER_INTERNAL_MERCH_ID": "xxxxxxxxxxxxx",
    "ENCRYPT_TO": "xxxxxxxxxxxxxxxxxxxx",
    "DOMAIN_LIST": [
      {
        "DOMAIN_NAME": "paymentpagedomain.com",
        "DOMAIN_STATUS": "ACTIVE"
      }
    ],
    "paymentRequest": {
      "countryCode": "US",
      "currencyCode": "USD",
      "merchantCapabilities": [
        "supports3DS"
      ],
      "supportedNetworks": [
        "visa",
        "MasterCard"
      ],
      "requiredBillingContactFields": [
        "postalAddress",
        "name",
        "phoneticName",
        "phone",
        "email"
      "total": {
        "label": ""
        "type": "final",
        "amount": 0
      },
    },
    "fetchUrl": " https://[[DOMAIN]]/apple-pay-services/api/session/create "
  }
</script>
```

3.4 Update the configuration with the specific payment/order details

Once you have the configuration stored in the merchantConfig parameter on your payment page, you can update the details which are specific to the purchase.

Example:

```
<script>
  merchantConfig.total.label = “Some purchase description total”;
  merchantConfig.total.amount = 49.95
</script>
```

3.5 Add the Apple Pay library to your payment page

You must add the Apple Pay library to your payment page to make calls to the Apple Pay API,
To do that, source the Apple Pay library script source in the page’s source html

Example:

```
<script src="https://applepay.cdn-apple.com/jsapi/v1/apple-pay-sdk.js"></script>
```

Note that this library will include functionality to validate the user/browser/device as eligible for Apple Pay.

3.6 Create a JS function to validate your domain

The Apple Pay JS API library will need to call a function that you define in order to perform a validation on the domain where your payment page is hosted. Note that all of this must load directly from your servers on this domain (no proxies, iframes, etc.)

Example:

```
<script>
 function validateMerchant(merchantConfig) {
   console.log("validateMerchant: ", merchantConfig.DOMAIN_LIST) const data = {
       method: "POST",
       headers: {
           "Content-type": "application/json",
           "Accept": "*/*"
       },
       body: JSON.stringify({
           "merchantIdentifier": merchantConfig.PARTNER_INTERNAL_MERCH_ID,
           "displayName": merchantConfig.DISPLAYNAME,
           "initiative": merchantConfig.INITIATIVE,
           "initiativeContext": merchantConfig.DOMAIN_LIST[0].DOMAIN_NAME
       })
   };
   return fetch(merchantConfig.fetchUrl, data);
}
<script>
```

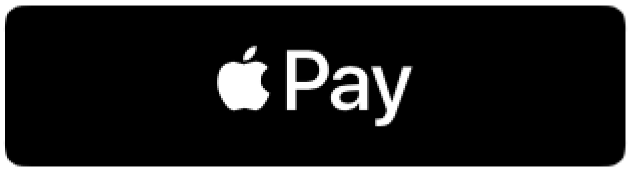
3.7 Present the Apple Pay button

Once the Apple Pay library is added, you can add an HTML element. Please note that it is required to name the element “apple-pay-button”.

Example:

```
<style>
.apple-pay-button {
   --apple-pay-button-width: 150px;
   --apple-pay-button-height: 30px;
   --apple-pay-button-border-radius: 3px;
   --apple-pay-button-padding: 0px 0px;
   --apple-pay-button-box-sizing: border-box;
}
</style>
```

The customer will see the Apple Pay button:



3.8 Create and start the Apple Pay session

After presenting the button, you must add an action to it which will start up the session where the browser will pop up a dialog for the user to select their card and approve the transaction.

For this, add an onClick event to the Apple Pay button. When referring to "paymentRequest", we are looking at the JSON object returned from APPLEPAYCONFIG. An example of this is included below

3.8.1 Handle the Authorized Payment

After the customer has authorized the payment via the Apple Pay overlay, Safari will return an object (Token) with the data needed to transfer to the gateway API to authorize a payment.

For this, you must create the session.onpaymentauthorized method to receive the token object. An example of this is included below

3.8.2 Handle a Canceled Payment or Error

You should also handle the case where the customer does not authorize the payment or in case an error is thrown.

For this, you must create the session.onCancel. An example of this is included below.

Here is an example of all of these elements described above:

```
<script>
  async function onApplePayClick() {
    // Check for valid ApplePay session
    if(!ApplePaySession){
      return;
    }
    // Get the paymentRequest data from the merchantConfig object
    const paymentRequest = this.merchantConfig.paymentRequest;
    // Validate the merchant.
    // Note that this.merchantConfig is defined in Step 3
    const session = new ApplePaySession(3, paymentRequest);
    session.onvalidatemerchant = async (event) => {
      const merchantSession = await
        this.validateMerchant(this.merchantConfig).then(res=>res.json());
      session.completeMerchantValidation(merchantSession);
    };
    // Handle the authorized payment
    // Note, this must come inside the onApplePayClick() function, and
    // before session.begin
    session.onpaymentauthorized = async (event) => {
      // Define ApplePayPaymentAuthorizationResult
      const result = {
        status: ApplePaySession.STATUS_SUCCESS,
      };
      var applePaymentToken = event.payment;
      var tokenEncoded = btoa(JSON.stringify(applePaymentToken));

      // Now that the payment is authorized in Apple Pay, you can
      // initiate a service call to the gateway passing the
      // applePaymentToken into the gateway API parameter: PMT_WALLET_CRYPTOGRAM
      //
      // Do that here, through your server

      // If the payment is declined at the processor, then you should
      // put an appropriate message in place to the customer
      // result.status = ApplePaySession.STATUS_FAILURE
      // Now we can gracefully complete the Apple Pay browser interaction
      session.completePayment(result);
    };
    // In case the customer cancels or in case of an unexpected error
    // Note, this must come inside the onApplePayClick() function, and
    // before session.begin
    session.oncancel = (event) => {
      // Payment canceled by WebKit
      // error handling
      console.log("Payment canceled by WebKit: "+JSON.stringify(event.error))
    }
    // now the user will interact to approve the purchase
    // Note, this must come at the END of the onApplePayClick()
    session.begin();
  }
</script>
```

3.9 Send the authorized payment data from the session to the gateway

Here is an example of a request to authorize payment with Apple Pay:

```
https://[domain]/payment/pmt_service.cfm?request_action=CCAUTHCAP&
li_count_1=1&li_prod_id_1=111&li_value_1=49.95req_username=GATEWAY
USER&req_password=GATEWAY
PASS&site_id=11111&request_response_format=JSON&request_api_version=4.6&request_cu
rrency=USD&PMT_WALLET=applepay &PMT_WALLET_CRYPTOGRAM=xxxxxxx
```

3.9.1 Important requirements for Apple Pay authorization requests

You must NOT include any of the following parameters, which will cause the gateway to reject your request:

```
PMT_NUMB
PMT_KEY
PMT_EXPIRY
TOKEN_GUID
PMT_ID
PMT_LAST4
PMT_ID_XTL
PMT_NUMB_COF
REQUEST_INITIATOR
```

Other gateway parameters can be included, but MUST match the values used in the Apple Pay session.

- Parameter li_value_1 must match the value of paymentRequest.total in the merchantConfig data.
- Parameter request_currency must match the value of paymentRequest.currencyCode in the merchantConfig data

3.10 Rebiling and Credentials on File

The Inovio response will return a CUST_ID and PMT_ID on a successful authorization using an Apple Pay token (in the same way that it does for normal transactions).

To authorize on the generated ApplePay payment record, the same CUST_ID and PMT_ID should be passed in on all subsequent rebills or unscheduled merchant initiated card-on-file transaction requests.

4 *Apple Pay Merchant Decryption*

4.1 Overview

This section discusses how to send credit card transactions to the Payment Gateway using Apple Pay that has been decrypted by the client. The Inovio Gateway platform supports tokens from 3rd parties in accordance with the EMVCo specifications. The integrated application is responsible for decoding the data-string produced by the digital wallet / e-wallet output and sending the appropriate data with the transaction request.

4.2 Request Parameters

Below is a list of transaction request paramaters that are needed to route third party tokens properly.

Field Name	Description	Additional Notes
WALLET_TOKEN	Field to tell Inovio that the following transaction is using a 3rd party wallet provider.	Field is always required. Acceptable values: ApplePay, GooglePay
WALLET_TAVV	Contains the cryptogram returned by the token provider	Field is always required.
WALLET_ECI	Contains the ECI response value returned by the Token Provider.	Variable Type: Numeric Max Length: 2
WALLET_TID	Contains the token providers token transaction identifier (TID).	Required to be sent with the transaction if returned by the Token Provider

Here is an example of a request to authorize payment with Apple Pay:

```
/payment/pmt_service.cfm?request_action=CCAUTHCAP&merch_acct_id=120603&site_id=100
103&li_prod_id_1=111205&pmt_numb=************1600%20&pmt_expiry=122029&pmt_key=123
&li_value_1=1.00&request_currency=USD&cust_fname=Gilberto&cust_lname=Gilberto&cust
_email=test%40test.com&bill_addr_country=US&bill_addr_state=CA&bill_addr_city=Los%
20Angeles&bill_addr=4022%20Randolph%20Street&bill_addr_zip=90401&XTL_IP=11.11.11.1
1&request_api_version=20&request_response_format=JSON&req_username=ellenscoffee%40
gmail.com&req_password=Password123456789&argdebug=1&WALLET_ECI=07&WALLET_TID=tyujv
cyug54321jksdkfjskjdfjk&WALLET_TAVV=%2F1QRHvYAB+88c3MOKPZ3MAACAAA%3D&WALLET_TOKEN=
applepay
```