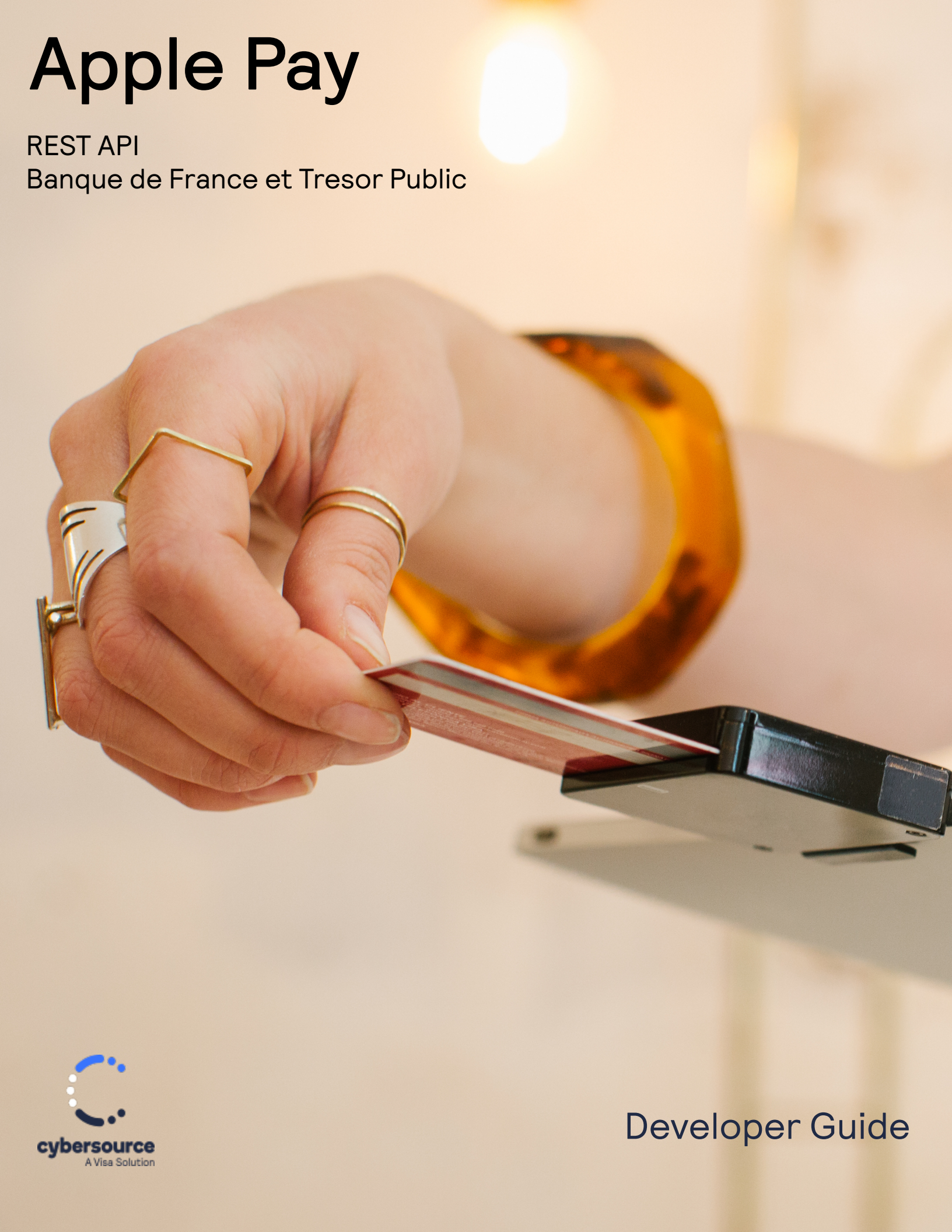
Apple Pay

REST API

Banque de France et Tresor Public



cybersource
A Visa Solution

Developer Guide

Cybersource Contact Information

For general information about our company, products, and services, go to https://www.cybersource.com.

For sales questions about any Cybersource service, email sales@cybersource.com or call 650-432-7350 or 888-330-2300 (toll free in the United States).

For support information about any Cybersource service, visit the Support Center: https://www.cybersource.com/support

Copyright

Restricted Rights Legends

Trademarks

Confidentiality Notice

This document is furnished to you solely in your capacity as a client of Cybersource and as a participant in the Visa payments system.

By accepting this document, you acknowledge that the information contained herein (the "Information") is confidential and subject to the confidentiality restrictions contained in Visa's operating regulations and/or other confidentiality agreements, which limity our use of the Information. You agree to keep the Information confidential and not to use the Information for any purpose other than its intended purpose and in your capacity as a customer of Cybersource or as a participant in the Visa payments system. The Information may only be disseminated within your organization on a need-to-know basis to enable your participation in the Visa payments system. Please be advised that the Information may constitute material non-public information under U.S. federal securities laws and that purchasing or selling securities of Visa Inc. while being aware of material non-public information would constitute a violation of applicable U.S. federal securities laws.

Revision

Version: 24.02

Contents

Recent Revisions to This Document

24.02

Clarified Apple Pay requirements. See *Requirements for Using Apple Pay* on page 7.

24.01

This revision contains only editorial changes and no technical updates.

23.04

This revision contains only editorial changes and no technical updates.

23.03

This revision contains only editorial changes and no technical updates.

23.02

This revision contains only editorial changes and no technical updates.

About This Guide

This section describes the audience and purpose of this guide and the conventions used in this guide.

Audience and Purpose

This document is written for merchants who want to use Apple Pay in an iOS application and use information from Apple to process payments through Cybersource. This document provides an overview for integrating Apple and Cybersource services into an order management system.

Conventions

The following special statement is used in this document:

> **Important**
>
> An Important statement contains information essential to successfully completing a task or learning a concept.

Customer Support

For support information about any service, visit the Support Center:
http://support.visaacceptance.com

Introduction to Apple Pay

You can use the Cybersource platform to process and manage Apple Pay transactions. Apple Pay is a digital payment solution that allows customers to maake secure and convenient transactions using their Apple devices. By adding Apple Pay to your web page, you can offer your customers a faster and easier way to pay online, without requiring them to enter their card details or shipping information. Apple Pay also reduces the exposure of sensitive payment data to your system, as the payment information is encrypted and tokenized by Apple.

Requirements for Using Apple Pay

In order to use the Cybersource platform to process Apple Pay transactions, you must have these things:

- A Cybersource account. If you do not already have a Cybersource account, contact your local Cybersource sales representative.
- A *live merchant account* with a *supported processor* to handle processing the transactions.
- An Admin or Team Agent user of the *Apple Pay Developer* account to initiate the transactions.

Important

Apple Pay relies on authorizations with payment network tokens. You can sign up for Apple Pay only when these statements are true:

- Your processor supports payment network tokens.
- Cybersource supports payment network tokens with your processor.

If one of these statements is not true, you must take one of these actions before you can sign up for Apple Pay:

- Obtain a new merchant account with a processor that supports payment network tokens.

- Wait until your processor supports payment network tokens.

Supported Card Types and Optional Features

This table explains which card types and optional features are supported for each processor.

Processor	Card Types	Optional Features
Banque de France et Tresor Public	Cartes Bancaires Mastercard Visa	Recurring payments

Payment Processors

Payment processors connect with acquirers. Before you can accept payments, you must register with a payment processor. An acquirer might require you to use a payment processor with an existing relationship with the acquirer.
Your payment processor assigns one or more merchant IDs (MIDs) to your business. These unique codes identify your business during payment transactions.

Apple Pay Integrations and Payload Decryption Methods

In response to a payment request, Apple Pay returns an encrypted payload that contains sensitive payment information. Two methods can be used to extract and decrypt the payment information, and both methods support Apple Pay in-app and web transactions.

- With Cybersource decryption, Cybersource manages the generation of the payment encryption key and the decryption of the Apple Pay response.

  Important

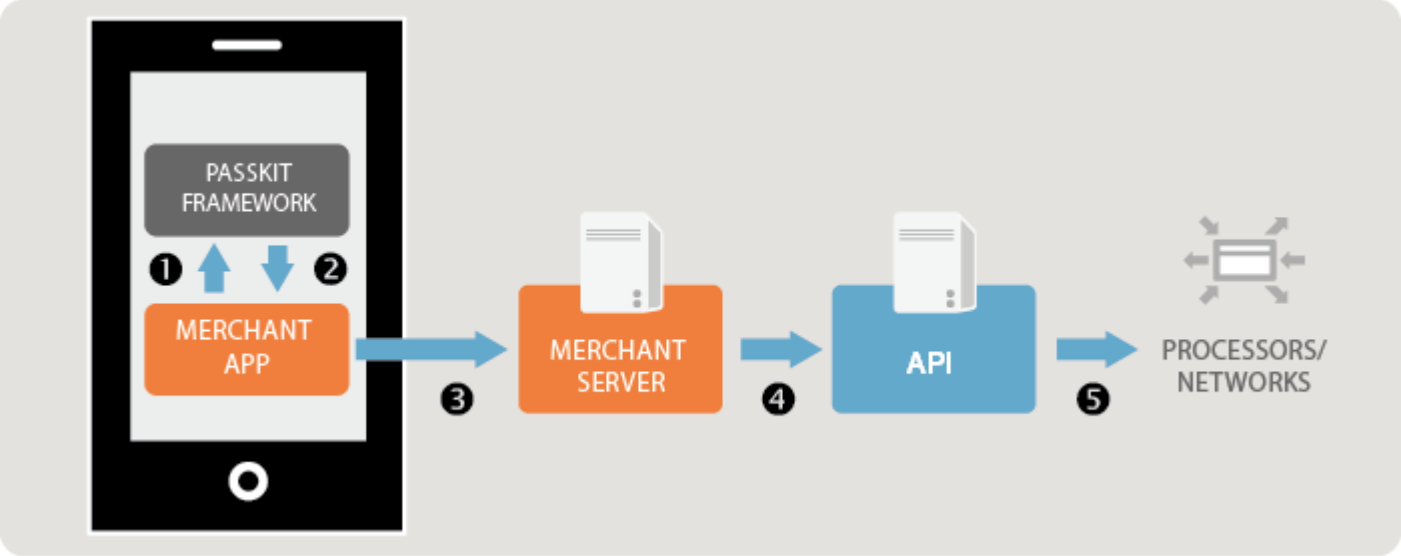
  This method reduces the exposure of sensitive payment data to your system.

- With merchant decryption, you manage both the payment encryption key generation and the decryption of the Apple Pay response.

The remainder of this section summarizes the flow of Apple Pay transaction processing for each decryption method.

Flow of Apple Pay In-App Transactions Processed Using Cybersource Decryption

This diagram illustrates the flow of in-app transaction processing using Cybersource decryption.



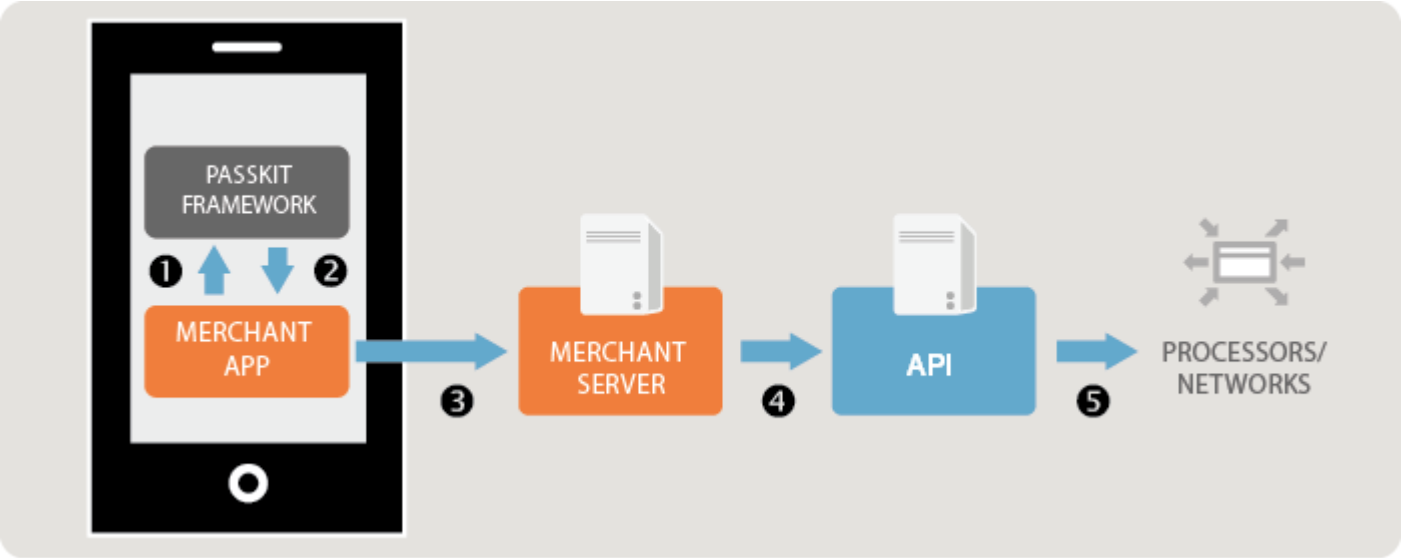
In-App Transaction Processing with Cybersource Decryption

1. When the customer chooses to pay with Apple Pay, you use the Apple PassKit Framework to request the encrypted payment data from Apple.
2. Apple uses the Secure Element to create a payment token and encrypt the token's payment data before it sends your application.
   - The payment token is the `PKPaymentToken` structure.
   - The payment data is the **paymentData** field of the `PKPaymentToken` structure.
3. You forward the encrypted payment data to your order management system.
4. Using the Cybersource API, you submit the authorization request. In the **paymentInformation.fluidData.value** field, include the Base64-encoded value that you obtained from the **paymentData** field of the `PKPaymentToken` structure.
5. Cybersource decrypts the payment data and forwards the information to the payment network, which includes your processor and the relevant payment card company.

> **Important**
>
> You must use the Business Center or one of the Cybersource API services to capture, credit, or void the authorization.

Flow of Apple Pay In-App Transactions Processed Using Merchant Decryption

This diagram illustrates the flow of in-app transaction processing using merchant decryption.



In-App Transaction Processing with Merchant Decryption

1. When the customer chooses to pay with Apple Pay, you use the Apple PassKit Framework to request the encrypted payment data from Apple.
2. Apple uses the Secure Element to create a payment token and encrypt the token's payment data before it sends your application.
   - The payment token is the `PKPaymentToken` structure.
   - The payment data is the **paymentData** field of the `PKPaymentToken` structure.
3. You forward the encrypted payment data to your order management system to decrypt. For information on decryption, see the *Payment Token Format Reference* article in the Apple Pay Developer Documentation.
4. Using the Cybersource API, you submit the authorization request and include the decrypted payment data.
5. Cybersource forwards the information to the payment network, which includes your processor and the relevant payment card company.

> **Important**
>
> You must use the Business Center or one of the Cybersource API services to capture, credit, or void the authorization.

Flow of Apple Pay Web Transactions Using Cybersource Decryption

1. When the customer chooses to pay with Apple Pay, you use the Apple Pay JavaScript to request the encrypted payment data from Apple.
2. Apple uses the Secure Element to create a payment token and encrypt the token's payment data before it sends your application using the `onpaymentauthorized` event handler.
   - The payment token is the `ApplePayPaymentToken` structure.

- The payment data is the **paymentData** field of the `ApplePayPaymentToken` structure.

For more information, see *Apple Pay on the Web Interactive Demo*.

3. You forward the encrypted payment data to your order management system.
4. Using the Cybersource API, you submit the authorization request. In the **paymentInformaton.fluidData.value** field, include the Base64-encoded value obtained from the **paymentData** field of the `ApplePayPaymentToken` structure.
5. Cybersource decrypts the payment data and forwards the information to the payment network, which includes your processor and the relevant payment card company.

> **Important**
>
> Use the Business Center or one of the Cybersource API services to capture, credit, or void the authorization.

Flow of Apple Pay Web Transactions Using Merchant Decryption

1. When the customer chooses to pay with Apple Pay, you use the Apple Pay JavaScript to request the encrypted payment data from Apple.
2. Apple uses the Secure Element to create a payment token and encrypt the token's payment data before it sends your application using the `onpaymentauthorized` event handler.

   - The payment token is the `ApplePayPaymentToken` structure.
   - The payment data is the **paymentData** field of the `ApplePayPaymentToken` structure.

   For more information, see *Apple Pay on the Web Interactive Demo*.
3. You forward the encrypted payment data to your order management system to decrypt. For information on decryption, see the *Payment Token Format Reference* article in the Apple Pay Developer Documentation.
4. Using the Cybersource API, you submit the authorization request and include the decrypted payment data. See *Authorization Service*.
5. Cybersource forwards the information to the payment network, which includes your processor and the relevant payment card company.

> **Important**
>
> Use the Business Center or one of the Cybersource API services to capture, credit, or void the authorization.

Configuring Apple Pay Processing

A successful Apple Pay response to a payment request returns an encrypted payload that contains sensitive payment information. The payment information is extracted and decrypted using Cybersource decryption or merchant decryption.
Before you can process Apple Pay transactions, you must complete these configuration tasks:

Generating and Loading a New Certificate Signing Request	For processing Apple Pay transactions using Cybersource decryption, you must first generate an Apple Pay encryption key on the business portal and load it into the Apple development portal. For merchant decryption, this configuration task is not needed.
Configuring Apple Pay Response Handling	After the payment token is received, the transaction is finalized when you extract and decrypt the payment data. Cybersource decryption and merchant decryption methods consist of different configuration steps.

Generating and Loading a New Certificate Signing Request

This task is required in order to process Apple Pay transactions using Cybersource decryption. It does not apply to merchant decryption.

Follow these steps in order to configure Apple Pay in the Cybersource Business Center:

1. Log in to the Business Center:
   - Test: *https://businesscentertest.cybersource.com*
   - Production: *https://businesscenter.cybersource.com*
2. On the left navigation panel, click the **Payment Configuration** icon.
3. Click **Digital Payment Solution**. The Digital Payments page appears. If you do not have the correct permissions enabled on your account, the Digital Payment Solution option does not appear on the left navigation panel.
4. Click **Configure**. The Apple Pay Registration panel opens.
5. Enter your Apple Merchant ID.
6. Click **Generate New CSR**.
7. To download your CSR, click the **Download** icon next to the key.
8. Follow your browser's instructions to save and open the file.
9. Complete the enrollment process by submitting your CSR to Apple.
10. For information about adding certificates to your Apple Merchant ID, refer to the Apple Pay PassKit:
    - In-App: *Setting Up Apple Pay*
    - Website: *Configuring Your Environment*
11. Test your software by following the steps in *Authorization Service* on page 16.

> **Important**
>
> If you are using a Cybersource test account, you must connect to the *Apple sandbox* tester account and not to the Apple production system.
> After you complete your testing, you must create a new CSR for the Cybersource production system, and you must use that CSR for the Apple production system. Until you perform these steps, you cannot enable payments in your iOS application or website.

12. Repeat Steps 1 through 11 with your Cybersource production account and the Apple production account.

Configuring Apple Pay Response Handling

Configure one of the following decryption methods that will handle the response payload of a successful Apple Pay transaction.

After the payment token is received, the transaction is finalized when you extract and decrypt the payment data.

- Cybersource Decryption:
  This method forwards the encrypted payment data to your order management system. Use the Cybersource API to submit the authorization request, and

include the Base64-encoded value obtained from the `paymentData` object in the **paymentInformaton.fluidData.value** field.
Example of Cybersourcedecryption:

```
session.onpaymentauthorized = function (event) {
   var paymentDataString = JSON.stringify(event.payment.token.paymentData);
   var paymentDataBase64 = btoa(paymentDataString);
}
```

- Merchant Decryption:
  This method forwards the encrypted payment data to your order management system to decrypt. Use the Cybersource API to submit the authorization request and include the decrypted payment data.
  For detailed information about decryption, see *Payment Token Format Reference* in the Apple Developer Center.

Processing Apple Pay Transactions

This section provides information about the different kinds of Apple Pay transactions that you can submit:

Authorizing a Payment Using Cybersource Decryption or Merchant Decryption	You can request the authorization service using these decryption methods: • Cybersource decryption: Within this integration, Cybersource facilitates all of the encryption key generation and decryption processes associated with implementing Apple Pay. This method reduces the exposure of the sensitive payment data to your system. • Merchant decryption: Within this integration, you manage all aspects of the Apple Pay integration, from generation of the payment encryption keys to decryption of the Apple Pay payload response. As a merchant, you submit the Apple Pay payment token and other payment information to Cybersource for processing.
Reversing an Authorization	This service uses the request ID returned from the previous authorization. An authorization reversal releases the hold that the authorization placed on the customer's credit card funds. Use this service to reverse an unnecessary or undesired authorization.
Capturing a Payment	This service uses the request ID returned from the previous authorization. The

	request ID links the capture to the authorization. Use this service to transfer funds from the customer's account to your bank. The transaction typically completes in two to four days.
Authorizing and Capturing a Payment	A bundled authorization and capture is called a sale. Request the authorization and capture services at the same time. Cybersource processes the capture immediately.

Authorization Service

This section provides the information that you need for requesting the authorization service using these methods which determine the request fields that are required to request the authorization service. In addition, different request fields are required in order to request the authorization service. The type of card used to process the transaction determines which request fields are used.
After you send the request, verify the response messages to make sure that the request was successful. A 200-level HTTP response code indicates success.
For information about response codes, see *Transaction Response Codes*.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Authorizations Using Cybersource Decryption for Mastercard

This section provides the information you need in order to process an authorization using Cybersource decryption for Mastercard.

Required Fields for Authorizing a Payment Using Cybersource Decryption for Mastercard

clientReferenceInformation.code

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.type	
paymentInformation.fluidData.descriptor	**Set this field to** `RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U`**.**
paymentInformaton.fluidData.value	**Set this field to the Base64-encoded value obtained from the** `paymentData` **property of the** `PKPaymentToken` **object.**
paymentInformation.tokenizedCard.transactionType	**Set this field to** `1`**.**
processingInformation.paymentSolution	**Set this field to** `001`**.**

Related Information

- *API Field Reference for the REST API*

REST Example: Cybersource Decryption and Mastercard

Request

```
{
  "clientReferenceInformation": {
    "code": "1234567890"
  },
  "processingInformation": {
    "paymentSolution": "001"
  },
  "paymentInformation": {
    "fluidData": {
      "value"="eyJkYXRhW5FINWZqVjfkak1NdVNSaE96dWF2ZGVyb2c9PSJ9",
      "descriptor": "RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U",
      "encoding": "Base64",
    },
   "tokenizedCard": {
      "type": "002",
      "transactionType": "1"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
```

```
      },
      "billTo": {
        "firstName": "Maya",
        "lastName": "Lee",
        "address1": "123 Main St",
        "locality": "SomeCity",
        "administrativeArea": "CA",
        "postalCode": "94404",
        "country": "US",
        "email": "maya.lee@email.world"
      }
    }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6234236182176225003004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/623423618217622500 3004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6234236182176225003004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1234567890"
  },
  "id": "623423618217622500 3004",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "expirationYear": "2031",
      "prefix": "128945",
      "expirationMonth": "12",
      "suffix": "2398",
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
```

```
  "processingInformation": {
    "paymentSolution": "001"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "75729760OPN67ZFV",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2021-06-11T15:00:18Z"
}
```


Authorizations Using Cybersource Decryption for Visa

This section provides the information you need in order to process an authorization using Cybersource decryption for Visa.

Required Fields for Authorizing a Payment Using Cybersource Decryption for Visa

clientReferenceInformation.code	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.type	
paymentInformation.fluidData.descriptor	**Set this field to** `Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U`.

paymentInformaton.fluidData.value	**Set this field to the Base64-encoded value obtained from the `paymentData` property of the `PKPaymentToken` object.**
paymentInformation.tokenizedCard.transactionType	**Set this field to `1`.**
processingInformation.paymentSolution	**Set this field to `001`.**

Related Information

- *API Field Reference for the REST API*

REST Example: Cybersource Decryption for Visa

Request

```
{
  "clientReferenceInformation": {
    "code": "1234567890"
  },
  "processingInformation": {
    "paymentSolution": "001"
  },
  "paymentInformation": {
    "fluidData": {
      "value"="eyJkYXRhW5FINWZqVjfkak1NdVNSaE96dWF2ZGVyb2c9PSJ9",
      "descriptor": "RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U",
      "encoding": "Base64",
    },
   "tokenizedCard": {
      "type": "001",
      "transactionType": "1"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    },
    "billTo": {
      "firstName": "Maya",
      "lastName": "Lee",
      "address1": "123 Main St",
      "locality": "SomeCity",
      "administrativeArea": "CA",
      "postalCode": "94404",
      "country": "US",
      "email": "maya.lee@email.world"
    }
  }
}
```

Response to a Successful Request

```
{
```

```
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6234236182176225003004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/623423618217622500 3004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6234236182176225003004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1234567890"
  },
  "id": "6234236182176225003004",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "expirationYear": "2031",
      "prefix": "411111",
      "expirationMonth": "12",
      "suffix": "1111",
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processingInformation": {
    "paymentSolution": "001"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "75729760OPN67ZFV",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2021-06-11T15:00:18Z"
}
```

Authorizations Using Merchant Decryption for Mastercard

This section provides the information you need in order to process an authorization using merchant decryption for Mastercard.

Required Fields for Authorizing a Payment Using Merchant Decryption for Mastercard

clientReferenceInformation.code	
consumerAuthenticationInformation.ecommerceIndicator	**Set this field to** `internet`**.**
consumerAuthenticationInformation.ucafAuthenticationData	**Set this field to the network token cryptogram.**
consumerAuthenticationInformation.ucafCollectionIndicator	**Set this field to** `2`**.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.tokenizedCard.cryptogram	**Set this field to the network token cryptogram.**
paymentInformation.tokenizedCard.expirationMonth	**Set this field to the value from the payment network token expiration month.**
paymentInformation.tokenizedCard.expirationYear	**Set this field to the value from the payment network token expiration year.**
paymentInformation.tokenizedCard.number	**Set this field to the payment network token value.**
paymentInformation.tokenizedCard.transactionType	**Set this field to** `1`**.**
processingInformation.paymentSolution	**Set this field to** `001`**.**

Related Information

- *API Field Reference for the REST API*

REST Example: Merchant Decryption and Mastercard

Request

```
{
  "clientReferenceInformation": {
    "code": "1234567890"
  },
  "processingInformation": {
    "paymentSolution": "001",
    "commerceIndicator": "spa"
  },
  "paymentInformation": {
   "tokenizedCard": {
      "number": "5432543254325432",
      "expirationMonth": "12",
      "expirationYear": "2031",
      "cryptogram": "ABCDEFabcdefABCDEFabcdef0987654321234567",

      "transactionType": "1",
      "type": "002"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    },
    "billTo": {
      "firstName": "Maya",
      "lastName": "Lee",
      "address1": "123 Main St",
      "locality": "SomeCity",
      "administrativeArea": "CA",
      "postalCode": "94404",
      "country": "US",
      "email": "maya.lee@email.world"
    }
  },
  "consumerAuthenticationInformation": {
    "ucafAuthenticationData": "ABCDEFabcdefABCDEFabcdef0987654321234567",
    "ucafCollectionIndicator": "2"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6234236182176225003004/reversals"
```

```
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/6234236182176225003004"
      },
      "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/6234236182176225003004/captures"
      }
    },
    "clientReferenceInformation": {
      "code": "1234567890"
    },
    "id": "6234236182176225003004",
    "orderInformation": {
      "amountDetails": {
        "authorizedAmount": "100.00",
        "currency": "USD"
      }
    },
    "paymentInformation": {
      "tokenizedCard": {
        "expirationYear": "2031",
        "prefix": "543254",
        "expirationMonth": "12",
        "suffix": "5432",
        "type": "002"
      },
      "card": {
        "type": "002"
      }
    },
    "pointOfSaleInformation": {
      "terminalId": "111111"
    },
    "processingInformation": {
      "paymentSolution": "001"
    },
    "processorInformation": {
      "approvalCode": "888888",
      "networkTransactionId": "123456789619999",
      "transactionId": "123456789619999",
      "responseCode": "100",
      "avs": {
        "code": "X",
        "codeRaw": "I1"
      }
    },
    "reconciliationId": "75729760OPN67ZFV",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2021-06-11T15:00:18Z"
}
```

Authorizations Using Merchant Decryption for Visa

This section provides the information you need in order to process an authorization using merchant decryption for Visa.

Required Fields for Authorizing a Payment Using Merchant Decryption for Visa

clientReferenceInformation.code	
consumerAuthenticationInformation.cavv	**Set this field to the network token cryptogram.**
consumerAuthenticationInformation.ecommerceIndicator	**Set this field to the ECI value contained in the Apple Pay response payload.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.tokenizedCard.cryptogram	**Set this field to the network token cryptogram.**
paymentInformation.tokenizedCard.expirationMonth	**Set this field to the value from the payment network token expiration month.**
paymentInformation.tokenizedCard.expirationYear	**Set this field to the value from the payment network token expiration year.**
paymentInformation.tokenizedCard.number	**Set this field to the payment network token value.**
paymentInformation.tokenizedCard.transactionType	**Set this field to `1`.**
processingInformation.paymentSolution	**Set this field to `001`.**

Related Information

- *API Field Reference for the REST API*

REST Example: Merchant Decryption and Visa

Request

```
{
  "clientReferenceInformation": {
    "code": "1234567890"
  },
  "processingInformation": {
    "paymentSolution": "001",
    "commerceIndicator": "internet"
  },
  "paymentInformation": {
   "tokenizedCard": {
      "number": "4111111111111111",
      "expirationMonth": "12",
      "expirationYear": "2031",
      "cryptogram": "AceY+igABPs3jdwNaDg3MAACAAA=",
      "transactionType": "1"
      "type": "001"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    },
    "billTo": {
      "firstName": "Maya",
      "lastName": "Lee",
      "address1": "123 Main St",
      "locality": "SomeCity",
      "administrativeArea": "CA",
      "postalCode": "94404",
      "country": "US",
      "email": "maya.lee@email.world"
    }
  },
  "consumerAuthenticationInformation": {
    "cavv": "AceY+igABPs3jdwNaDg3MAACAAA="
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6234236182176225003004/reversals"
    },
    "self": {
```

```
        "method": "GET",
        "href": "/pts/v2/payments/623423618217622500300 4"
      },
      "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/623423618217622500300 4/captures"
      }
    },
    "clientReferenceInformation": {
      "code": "123456789 0"
    },
    "id": "623423618217622500300 4",
    "orderInformation": {
      "amountDetails": {
        "authorizedAmount": "100.00",
        "currency": "USD"
      }
    },
    "paymentInformation": {
      "tokenizedCard": {
        "expirationYear": "2031",
        "prefix": "411111",
        "expirationMonth": "12",

        "suffix": "1111",
        "type": "001"
      },
      "card": {
        "type": "001"
      }
    },
    "pointOfSaleInformation": {
      "terminalId": "111111"
    },
    "processingInformation": {
      "paymentSolution": "001"
    },
    "processorInformation": {
      "approvalCode": "888888",
      "networkTransactionId": "123456789619999",
      "transactionId": "123456789619999",
      "responseCode": "100",
      "avs": {
        "code": "X",
        "codeRaw": "I1"
      }
    },
    "reconciliationId": "75729760OPN67ZFV",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2021-06-11T15:00:18Z"
}
```

Captures

This section provides the information you need in order to capture an authorized transaction.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments/{id}/captures
Test: POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures
The {id} is the transaction ID returned in the authorization response.

Required Fields for Capturing an Authorization

Use these required fields for capturing an authorization.

clientReferenceInformation.code	**This field value maps from the original authorization, sale, or credit transaction.**
clientReferenceInformation.partner.thirdPartyCertificationNumber	**Cybersource provides the value for this field.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	

REST Example: Capturing an Authorization

Request

```
{
  "clientReferenceInformation": {
    "code": "ABC123",
    "partner": {
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "EUR"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/captures/6662994431376681303954/voids"
    },
    "self": {
```

```
        "method": "GET",
        "href": "/pts/v2/captures/6662994431376681303954"
      }
    },
    "clientReferenceInformation": {
      "code": "1666299443215"
    },
    "id": "6662994431376681303954",
    "orderInformation": {
      "amountDetails": {
        "totalAmount": "100.00",
        "currency": "EUR"
      }
    },
    "reconciliationId": "66535942B9CGT52U",
    "status": "PENDING",
    "submitTimeUtc": "2022-10-20T20:57:23Z"
}
```


Sales

This section provides the information you need in order to process a sale transactions. A sale combines an authorization and a capture into a single transaction.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing a Sale

Use these required fields for processing a sale.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.capture	**Set the value to `true`.**

Related Information

- *API field reference guide for the REST API*

REST Example: Processing a Sale

Request

```
{
 "processingInformation": {
  "capture": true
 },
 "orderInformation" : {
  "billTo" : {
  "country" : "US",
  "lastName" : "VDP",
  "address1" : "201 S. Division St.",
  "postalCode" : "48104-2201",
  "locality" : "Ann Arbor",
  "administrativeArea" : "MI",
  "firstName" : "RTS",
  "email" : "test@cybs.com"
 },
  "amountDetails" : {
   "totalAmount" : "100.00",
   "currency" : "usd"
  }
 },
 "paymentInformation" : {
  "card" : {
   "expirationYear" : "2031",
   "number" : "4111111111111111",
   "expirationMonth" : "12",
   "type" : "001
  }
 }
}
```

Response to a Successful Request

Most processors do not return all of the fields that are shown in this example.

```
{
 "_links" : {
  "void" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6485004068966546103093/voids"
  },
  "self" : {
   "method" : "GET",
```

```
      "href" : "/pts/v2/payments/6485004068966546103093"
    }
  },
  "clientReferenceInformation" : {
    "code" : "RTS-Auth"
  },
  "id" : "6485004068966546103093",
  "orderInformation" : {
    "amountDetails" : {
      "totalAmount" : "100.00",
      "authorizedAmount" : "100.00",
      "currency" : "usd"
    }
  },
  "paymentAccountInformation" : {
    "card" : {
      "type" : "001"
    }
  },
  "paymentInformation" : {
    "tokenizedCard" : {
      "type" : "001"
    },
    "card" : {
      "type" : "001"
    }
  },
  "processorInformation" : {
    "systemTraceAuditNumber" : "841109",
    "approvalCode" : "831000",
    "merchantAdvice" : {
      "code" : "01",
      "codeRaw" : "M001"
    },
    "responseDetails" : "ABC",
    "networkTransactionId" : "016153570198200",
    "retrievalReferenceNumber" : "208720841109",
    "consumerAuthenticationResponse" : {
      "code" : "2",
      "codeRaw" : "2"
    },
    "transactionId" : "016153570198200",
    "responseCode" : "00",
    "avs" : {
      "code" : "Y",
      "codeRaw" : "Y"
    }
  },
  "reconciliationId" : "6485004068966546103093",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2022-03-28T20:46:47Z"
}
```

Searching for Apple Pay Transactions

Use the Transaction Search page in the Business Center to identify Apple Pay transactions. You can search for transactions by date, application type, customer name, and other transaction identifiers.
For information about the Transaction Request Report, see the Business Center *Reporting Developer Guide*.