

## Apple Pay

You can also accept payments through Apple Pay via [Checkout](#) (Hosted Payments Page - HPP).

Follow the integration steps below to process transactions via Verifone eComm API.

### Initiate a wallet payment using ApplePay

Check also our [API](#) documentation.

1. Enable ApplePay via Verifone's Hosted Checkout or via Direct API, as per the instructions from the [Apple Pay | Online payments | Verifone Developer Portal](#) documentation.

To enable the ApplePay wallet with Mobile SDKs check the [Apple Pay | Online payments | Verifone Developer Portal](#) information.

2. Generate a unique `merchant_reference`.

The `merchant_reference` needs to be unique to identify the shopper when they are redirected to your server by Apple Pay. In the next step, you will be creating a transaction through the API, the transaction will return an id that needs to be stored safely with the reference. When the shopper returns, you can use this reference to confirm either through the webhook or through the GET transaction API call if the shopper has successfully completed the transaction.

3. Required fields.

Parameters	Description
<code>payment_provider_contract</code>	In the <a href="#">Payment Provider Contracts</a> section in Verifone Central, set the <i>Payment Type</i> to <i>Apple Pay for web</i> , select your contract and copy the Payment Provider Contract ID. Note this value is different in Sandbox and in Production.
<code>amount</code>	Amount of the transaction.
<code>merchant_reference</code>	Unique UUID you generate and can link the transaction to when the customer returns.
<code>currency_code</code>	More on the all currencies supported here: <a href="#">Verifone eCommerce API   Verifone Developer Portal</a> .
<code>wallet_type</code>	string Enum: "APPLE_PAY"
<code>wallet_payload</code>	object The encrypted payload object provided by the Wallet on the frontend.

### Start a payment session for Apple Pay wallet transactions

The *Create a merchant payment session* API documentation can be found [here](#).

## Create a merchant payment session

Parameters	Description
<code>validation_url</code>	string The URL pointing to the Apple Pay validation location.
<code>domain</code>	string The domain from which the payment request will be initiated.
<code>payment_provider_contract</code>	string <uuid-flexible> The identifier of payment provider contract you want to process the transaction request with

## Web Integration

An Apple Pay web integration consists of implementing both client side and server side components. You will need to implement the following:

- Create your Apple Pay button and initialize your Apple Pay session when clicked.
  - [https://developer.apple.com/documentation/apple\\_pay\\_on\\_the\\_web/displaying\\_apple\\_pay\\_buttons\\_using\\_javascript](https://developer.apple.com/documentation/apple_pay_on_the_web/displaying_apple_pay_buttons_using_javascript)
- Initiate the merchant session by calling the on the `validateMerchant` event and provide the response to the `session.completeMerchantValidation` callback.
  - <https://verifone.cloud/api-catalog/verifone-ecommerce-api#operation/validationUrl>
- Use the Apple Pay token to make a wallet transaction API call and complete the Apple Pay session based on the response.
  - <https://verifone.cloud/api-catalog/verifone-ecommerce-api#operation/walletTransaction>

## Code sample

```
const ApplePaymentOptions = {
  countryCode: "US",
  currencyCode: "USD",
  merchantCapabilities: ["supports3DS"],
  supportedNetworks: ["visa", "masterCard", "amex", "discover"],
  total: {
    label: "Demo (Card is not charged)",
    type: "final",
    amount: "1.99",
  },
};

const canMakePayments =
  window.ApplePaySession && window.ApplePaySession.canMakePayments();
let session;
function handleApplePayRequest(formValues) {
  if (canMakePayments) {
    session = new window.ApplePaySession(3, ApplePaymentOptions);
    session.onpaymentauthorized = makeTransaction(formValues);
    session.addEventListener("validateMerchant", validateMerchant);
  }
}
```

```
session.begin();
console.log("INFO - BEGIN SESSION...");
}
function validateMerchant(e) {
// Pass the validation URL and domain to your server to make the wallet transaction API call
fetch(`YOUR_ENDPOINT`, {
method: "POST",
headers: REQUEST_HEADERS,
body: JSON.stringify({
validation_url: e.validationURL,
domain: window.location.hostname
}),
})
.then((res) => res.json())
.then((sessionObject) => {
session.completeMerchantValidation(sessionObject);
})
.catch((err) => console.log("ERROR - MERCHANT VALIDATION: ", err));
}
function makeTransaction(formValues) {
return (e) => {
const applePaymentObject = e.payment;
const payload = {
token: applePaymentObject.token.paymentData,
};
// Pass the apple pay token to your server to make the wallet transaction API call
fetch(`YOUR_ENDPOINT`, {
method: "POST",
headers: REQUEST_HEADERS,
body: JSON.stringify(payload),
})
.then((res) => res.json())
.then((res) => {
// Check the response status and complete the payment session
if (res.status === 'AUTHORIZED') {
session.completePayment(0);
}
})
.catch((err) => console.log("ERROR: ", err));
};
}
```