



<https://github.com/chrismorata/apple-pay-js>

What is Apple Pay JS?

Requirements

- <https://developer.apple.com/documentation/applepayjs>
- Must have an Apple Developer account
- Any page that uses Apple Pay must be served over HTTPS
- Must comply with Apple Pay guidelines

Guidelines

Requirements

- Your site can't violate any laws or engage in fraud
- Your site doesn't sell:
 - cigarettes or tobacco products
 - firearms, weapons, or ammunition
 - illegal drugs or non-legally prescribed controlled substances
 - items that create consumer safety risks
 - items that are intended to be used to engage in illegal activities
 - pornography
 - counterfeit or stolen goods

Requirements

- Your site doesn't promote hate, violence, or intolerance
- Users can't purchase currency using Apple Pay
- Violates intellectual property or privacy of someone
- Personal fundraising or nonprofit donations (unless approved by Apple)
- Shows Apple in a false or derogatory light

Design Guidelines

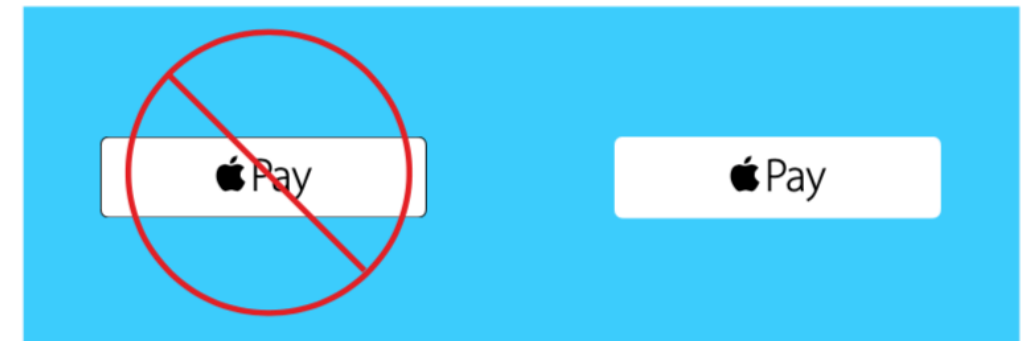
<https://developer.apple.com/apple-pay/Apple-Pay-Identity-Guidelines.pdf>

Apple Pay Buttons

Avoid Mistakes



Do not use the Apple logo alone.



On a saturated background color, use the white button without the outline rule.



Do not use the Apple Pay logo on its own within an app or website except as described for bank or card issuers on page 3, "Apple Pay logo."



On a black or very dark background, use the white button.



Integrating Apple Pay JS

JavaScript Promises

- JS is single threaded
- A promise is an object that may produce a single value some time in the future, write asynchronous code in a synchronous way
- Chain events together vs. callback hell
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises

Payment Request API

- Web standard for processing payments across systems, merchants, and platforms
- Outlines accepted payment methods, transaction details, shipping and contact info, line items, etc.
- Same structure for _____ Pay
- Google Pay just launched!

First things first.

HTML

```
<button class="apple-pay-button hidden"></button>
```

CSS

```
.apple-pay-button {  
    display: inline-block;  
    cursor: pointer;  
    -webkit-appearance: -apple-pay-button;  
}
```

First things first.

JAVASCRIPT

```
if ( window.ApplePaySession ) {  
  
}
```

Configure Your Payment Request

```
var PaymentRequest = {  
  currencyCode : 'USD',  
  countryCode : 'US',  
  requiredBillingContactFields : ['postalAddress'],  
  requiredShippingContactFields : ['name', 'email'],  
  supportedNetworks :  
    ['amex', 'discover', 'masterCard', 'visa'],  
  merchantCapabilities :  
    ['supportsCredit', 'supportsDebit', 'supports3DS'],  
  total : {  
    label : 'Total',  
    amount : 10.00  
  }  
};
```

Configure Your Payment Request

```
// use the merchant provided by Apple in your developer
account
var promise =
ApplePaySession.canMakePaymentsWithActiveCard('merchant.we
bsite.com');

// verify that Apple Pay is accepted
promise.then(function(canMakePayments){
    if ( canMakePayments ) {
        $(' .js-apple-pay-button').removeClass('hidden');
    } else {
        console.log('Apple Pay is not supported. ');
    }
});
```

Configure Your Payment Request

```
$('.js-apple-pay-button').on('click', function(e)
{
    // create Apple Pay session
    var PaymentSession = new ApplePaySession(2,
    PaymentRequest);

    // let's do some stuff!
});
```

Performing a Payment

```
PaymentSession.onvalidatemerchant = function(event){
    var promise = new Promise(function(resolve, reject){
        // AJAX call to backend to validate merchant goes here
        // response should be JSON-encoded
        $.ajax({
            url : 'path/to/merchantvalidation',
            success : function(response){
                resolve(JSON.parse(response));
            }
        });
    });

    // once the merchant is validated, the payment sheet is
    // displayed
    promise.then(function(merchantSession){
        PaymentSession.completeMerchantValidation(merchantSession);
    });
};
```


Performing a Payment

```
PaymentSession.onvalidatemerchant = function(e){
    var promise = new Promise(function(resolve, reject){
        // AJAX call to backend to validate merchant goes here
        // response should be JSON-encoded
        $.ajax({
            url : 'path/to/merchantvalidation?url=' + e.validationURL,
            success : function(response){
                resolve(JSON.parse(response));
            }
        });
    });

    // once the merchant is validated, the payment sheet is displayed
    promise.then(function(merchantSession){
        PaymentSession.completeMerchantValidation(merchantSession);
    });
};
```

Performing a Payment

```
// payment method selection
PaymentSession.onpaymentmethodselected = function(event){
    PaymentSession.completePaymentMethodSelection({ type:
        'final', label: 'Total', amount: 10.00 }, [{ type: 'final',
        label: 'Subtotal', amount: 10.00 }]);
};

// payment method authorization
// once payment is authorized, it will return an object that
// contains the payment token for your processor
PaymentSession.onpaymentauthorized = function(event){
    // add some logic here to work with your payment processor

    // complete payment session
    PaymentSession.completePayment(ApplePaySession.STATUS_SUCCESS
    );
};
```

Last, but not least.

```
PaymentSession.begin();
```

Working with a Payment Processor

<https://stripe.com/apple-pay>

Example