

Browser Payment Request API

Yutthana (Yut) - Tinn (Micky)

Growth Session #16 - July 12-13 2018

What is Web Payment Request API?

- The Payment Request API is a W3C standard candidate that is meant to **eliminate** checkout forms.
- The Payment Request API is designed to be vendor-agnostic, meaning it does not require use of a particular payment system.
- The Payment Request API is an open and cross-browser standard that replaces traditional checkout flows by allowing merchants to request and accept any payment in a single API.

The Payment Transaction Process



Site calls
PaymentRequest



Browser UI:
collect payment



Browser passes
response to site



Payment app

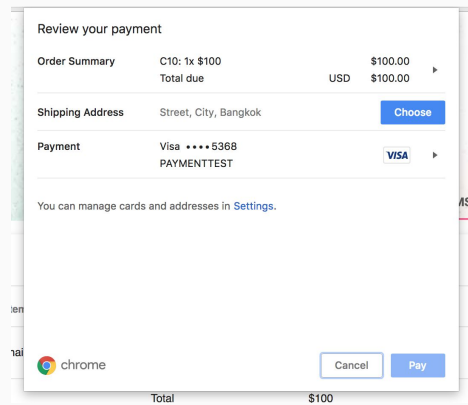
Feature Detection and Fallback



```
if (window.PaymentRequest) {  
  // Use Payment Request API  
} else {  
  // Fallback to traditional checkout  
  window.location.href = '/checkout/traditional';  
}
```

Feature Detection and Fallback

Browser supported



Review your payment

Order Summary	C10: 1x \$100	\$100.00
	Total due	USD \$100.00

Shipping Address Street, City, Bangkok [Choose](#)

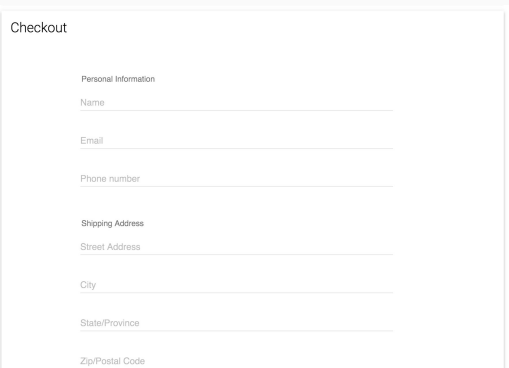
Payment Visa **** 5368 PAYMENTTEST [VISA](#)

You can manage cards and addresses in [Settings](#).

chrome [Cancel](#) [Pay](#)

Total \$100

Fallback



Checkout

Personal Information

Name

Email

Phone number

Shipping Address

Street Address


City

State/Province

Zip/Postal Code

```
if (window.PaymentRequest) {  
  // Use Payment Request API  
} else {  
  // Fallback to traditional checkout  
  window.location.href = '/checkout/traditional';  
}
```

Create Payment Request with PaymentRequest constructor



```
const payment = new PaymentRequest(  
  supportedPaymentMethods,  
  paymentDetails,  
  options  
);
```

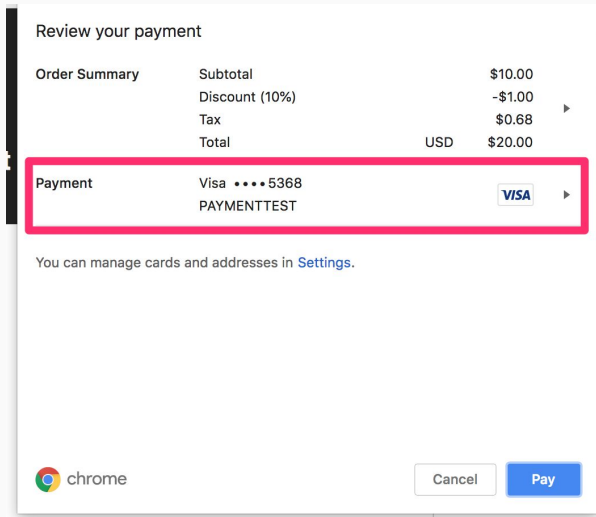
The parameters of PaymentRequest constructor

The `PaymentRequest` constructor requires **3** parameters to create an instance:

- `supportedPaymentMethods`
- `paymentDetails`
- `options`

The supportedPaymentMethods param

The [supportedPaymentMethods](#) defines which forms of payment you can accept; for example, you may only accept 'visa' and 'mastercard'.



Review your payment

Order Summary		
Subtotal		\$10.00
Discount (10%)		-\$1.00
Tax		\$0.68
Total	USD	\$20.00

Payment

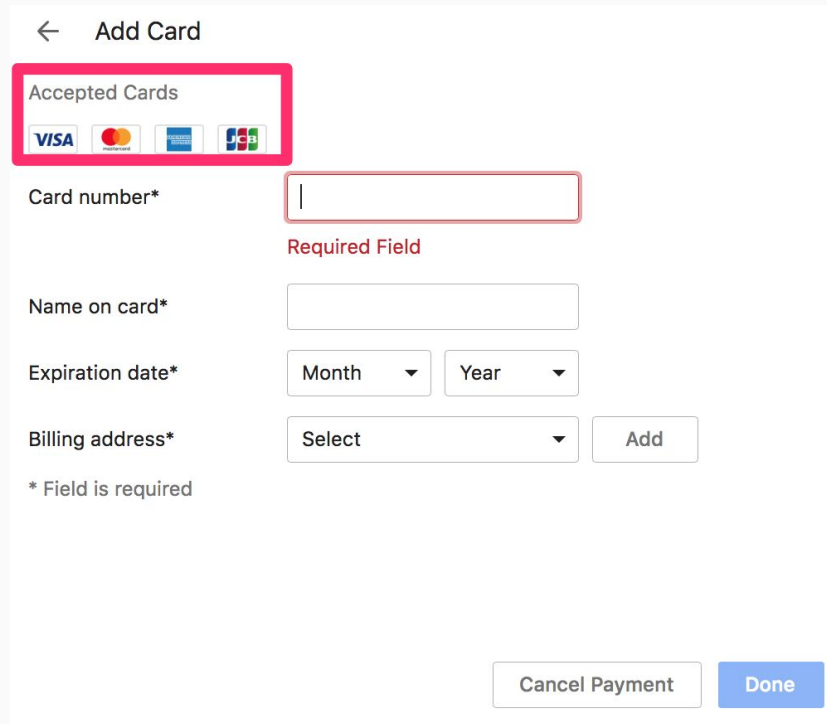
Visa ****5368
PAYMENTTEST

VISA

You can manage cards and addresses in [Settings](#).

chrome

Cancel Pay



← Add Card

Accepted Cards

VISA Mastercard American Express JCB

Card number*

Required Field

Name on card*

Expiration date* Month Year

Billing address* Select

* Field is required

Cancel Payment Done

The `supportedPaymentMethods` param

- This param must supply an array of objects indicating your supported payment methods where each payment method must include a `supportedMethods` parameter that identifies the payment method.
- Each object can contain an optional data object.

```
const supportedPaymentMethods = [{
  supportedMethods: 'name-of-payment-method',
  data: {
    // Optional data for this payment method
  }
}];

new PaymentRequest(supportedPaymentMethods, paymentDetails, options);
```

The paymentDetails param

The [paymentDetails](#) argument defines the total and display items.


Review your payment

Order Summary


Subtotal		\$10.00
Discount (10%)		-\$1.00
Tax		\$0.68
Total	USD	\$0.00

Payment

Visa ••••5368
PAYMENTTEST



You can manage cards and addresses in [Settings](#).

 chrome

Cancel

Pay

The `paymentDetails` param

- This param is an object.
- This object contains the total for the order and an optional array of display items

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It contains a JavaScript code snippet that defines a constant `paymentDetails` as an object. The object has a `total` property, which is itself an object containing `label` (set to 'Total'), `amount` (an object with `currency` set to 'USD' and `value` set to '0'), and an empty array for `displayItems`.

```
const paymentDetails = {  
  total: {  
    label: 'Total',  
    amount: {  
      currency: 'USD',  
      value: '0',  
    },  
    displayItems: [],  
  },  
};
```

The `options` param

The [options](#) is an object used to request additional information from the user; for example, you can request the payer's name, email address, and phone number.

Review your payment


Order Summary	Subtotal	\$10.00
	Discount (10%)	-\$1.00
	Tax	\$0.68
	Total	USD \$0.00

Payment

Visa ****5368
PAYMENTTEST

Contact Info Name, +66 12345678 **Choose**

You can manage cards and addresses in [Settings](#).

 chrome

Cancel **Pay**



← Add Contact Info

Name*
Required Field

Phone Number*


Email*

* Field is required

Cancel Payment **Done**

The options param

- This params is an object.
- This object should be used to define any additional information you require from the user



```
const options = {  
  requestPayerName: true,  
  requestPayerPhone: true,  
  requestPayerEmail: true,  
};
```

Responding to `PaymentRequest.show()`

After users fill in the required fields and select 'Pay', at which point your app will need to "complete" the transaction.

```
paymentRequest.show()  
.then((paymentResponse) => {  
  return validatePaymentWithBackend(paymentResponse)  
  .then((success) => {  
    if (success) {  
      return paymentResponse.complete('success');  
    } else {  
      return paymentResponse.complete('fail');  
    }  
  });  
})  
.catch((err) => {  
  console.log('Payment failed');  
});
```

paymentResponse

```
{
  "requestId": "7943e067-d14a-4ac8-9cfb-a73885c051aa",
  "methodName": "basic-card",
  "details": {
    "billingAddress": {
      "addressLine": [
        "Street"
      ],
      "city": "City",
      "country": "TH",
      "dependentLocality": "",
      "languageCode": "th",
      "organization": "",
      "phone": "+6612345678",
      "postalCode": "",
      "recipient": "Name",
      "region": "Bangkok",
      "sortingCode": ""
    },
    "cardNumber": "4778219765655368",
    "cardSecurityCode": "111",
    "cardholderName": "PAYMENTTEST",
    "expiryMonth": "03",
    "expiryYear": "2020"
  },
  "shippingAddress": null,
  "shippingOption": null,
  "payerName": "Name",
  "payerEmail": "Email@mail.com",
  "payerPhone": "+6612345678"
}
```

Thanks!

Contact Nimbl3

hello@nimbl3.com

399 Sukhumvit Road, Interchange 21
Klongtoey nua, Wattana
Bangkok 10110

28C Stanley St,
Singapore 068737

20th Floor, Central Tower
28 Queen's Road
Central, Hong Kong

nimbl3.com

