Bringing Easy and Fast Checkout with Payment Request API

By Eiji Kitamura
Developer Advocate in Tokyo

It's no surprise that the majority of online shopping is happening on mobile devices these days. But did you know that 66% of mobile purchases are made through websites rather than apps? Unfortunately though, conversion rate on mobile websites is only 33% of that on desktop. We need to fix this.

Chrome 53 for Android (desktop to be supported in the future) introduces a new API called Payment Request - a new approach for developers to eliminate checkout forms and improve user's payment experience from the ground up.

Introducing Payment Request API

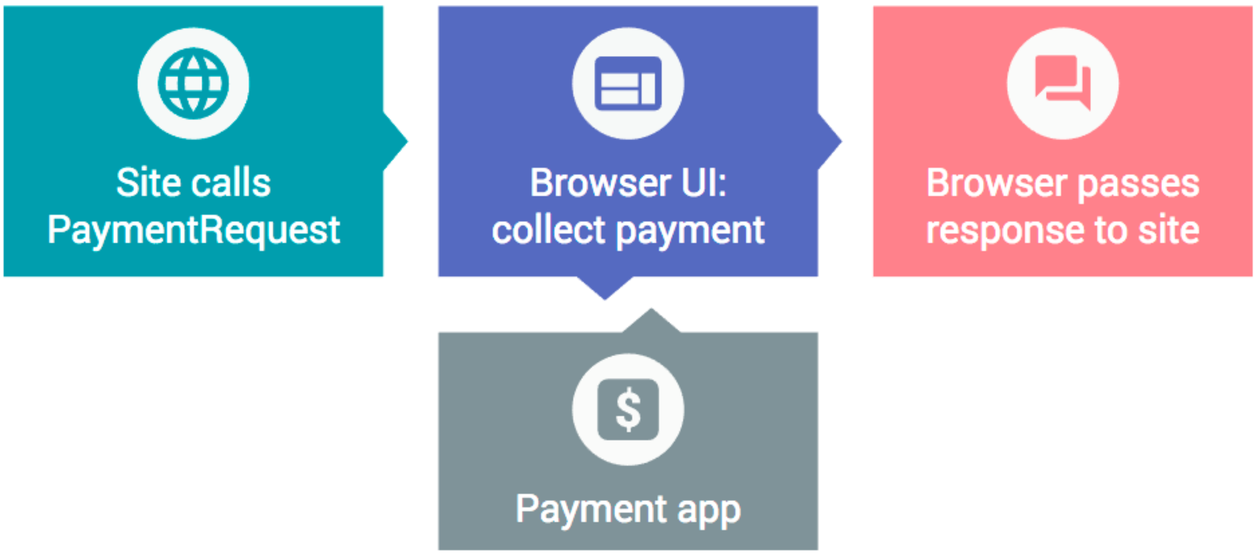
Payment Request is a new API for the open web that makes checkout flows easier, faster and consistent on shopping sites.

- Provides a native user interface for users to select or add a payment method, a shipping address and a shipping option in an easy, fast and consistent way.
- Provides standardized imperative APIs for developers to obtain user's payment preferences in a consistent format.

How the Payment Request API works

Let's peek at how Payment Request API works in some code. Here's a minimal example that collects a user's credit card information and submits it to a server.

```
function onBuyClicked() {
  if (!window.PaymentRequest) {
    // PaymentRequest API is not available. Forwarding to
    // legacy form based experience.
    location.href = '/checkout';
    return;
  }

  // Supported payment methods
  var supportedInstruments = [{
      supportedMethods: ['basic-card']
      data: {
        supportedNetworks: [
          'visa', 'mastercard', 'amex', 'discover',
          'diners', 'jcb', 'unionpay'
        ]
      }
  }];

  // Checkout details
  var details = {
    displayItems: [{
      label: 'Original donation amount',
      amount: { currency: 'USD', value: '65.00' }
    }, {
      label: 'Friends and family discount',
      amount: { currency: 'USD', value: '-10.00' }
    }],
    total: {
      label: 'Total due',
      amount: { currency: 'USD', value : '55.00' }
    }
  };

  // 1. Create a `PaymentRequest` instance
  var request = new PaymentRequest(supportedInstruments, details);

  // 2. Show the native UI with `.show()`
  request.show()
  // 3. Process the payment
  .then(result => {
    // POST the payment information to the server
    return fetch('/pay', {
      method: 'POST',
      credentials: 'include',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(result.toJSON())
    }).then(response => {
      // 4. Display payment results
      if (response.status === 200) {
        // Payment successful
        return result.complete('success');
      } else {
        // Payment failure
        return result.complete('fail');
      }
    }).catch(() => {
      return result.complete('fail');
    });
  });
}

document.querySelector('#start').addEventListener('click', onBuyClicked);
```


Site calls PaymentRequest
Browser UI: collect payment
Browser passes response to site
Payment app

1. Create a PaymentRequest instance

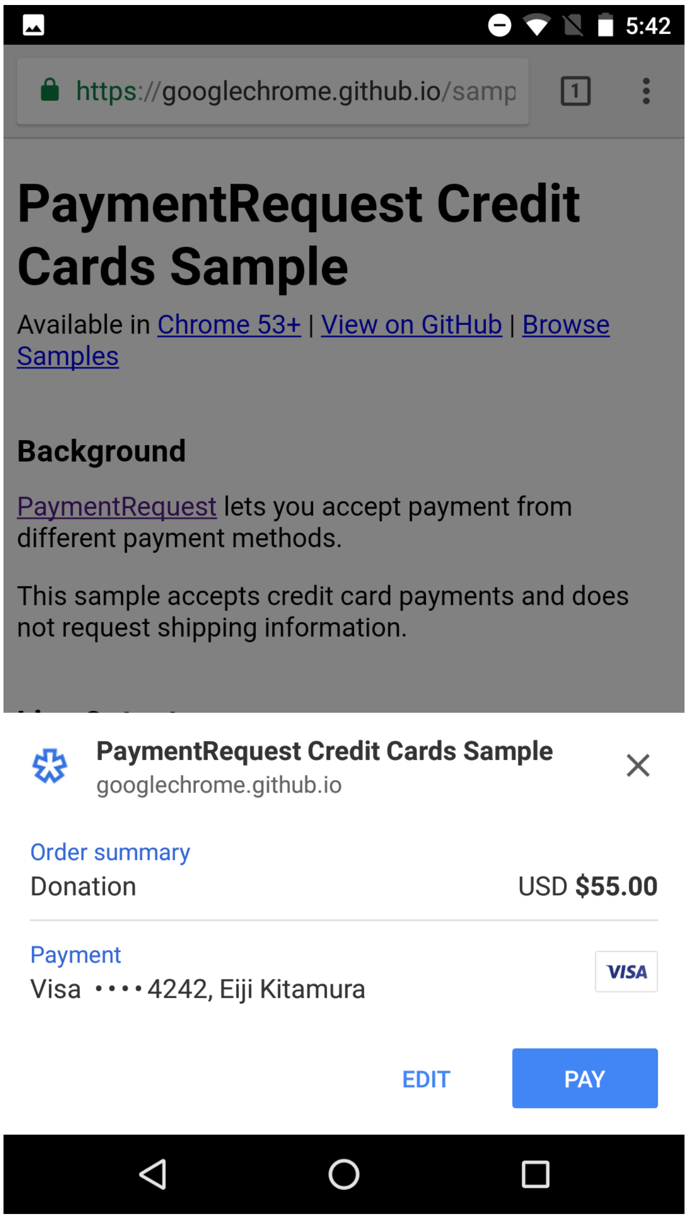
When a user taps on "Checkout", start a payment procedure by instantiating **`PaymentRequest`**.

```
var request = new PaymentRequest(supportedInstruments, details);
```

2. Show the native UI with .show()

Show the native payment UI with `show()`. Within this UI, a user can determine a payment method already stored in the browser or add a new one.

```
request.show()
.then(payment => {
  // pressed "Pay"
});
```


5:42

https://googlechrome.github.io/samp

PaymentRequest Credit Cards Sample

Available in Chrome 53+ | View on GitHub | Browse Samples

Background

PaymentRequest lets you accept payment from different payment methods.

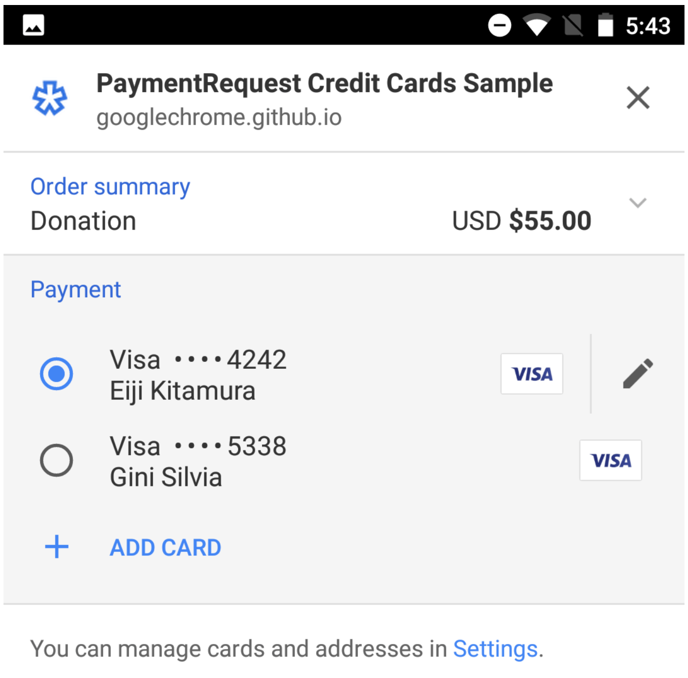
This sample accepts credit card payments and does not request shipping information.

PaymentRequest Credit Cards Sample
googlechrome.github.io

Order summary
Donation USD $55.00

Payment
Visa ••••4242, Eiji Kitamura

EDIT PAY


5:43

PaymentRequest Credit Cards Sample
googlechrome.github.io

Order summary
Donation USD $55.00

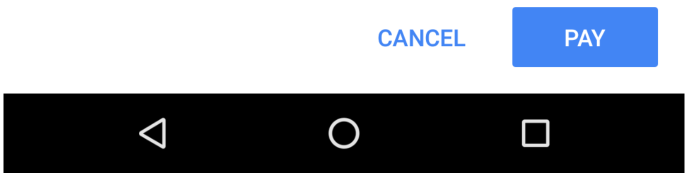
Payment

[✓ Visa ••••4242
Eiji Kitamura]

Visa ••••5338
Gini Silvia

+ ADD CARD

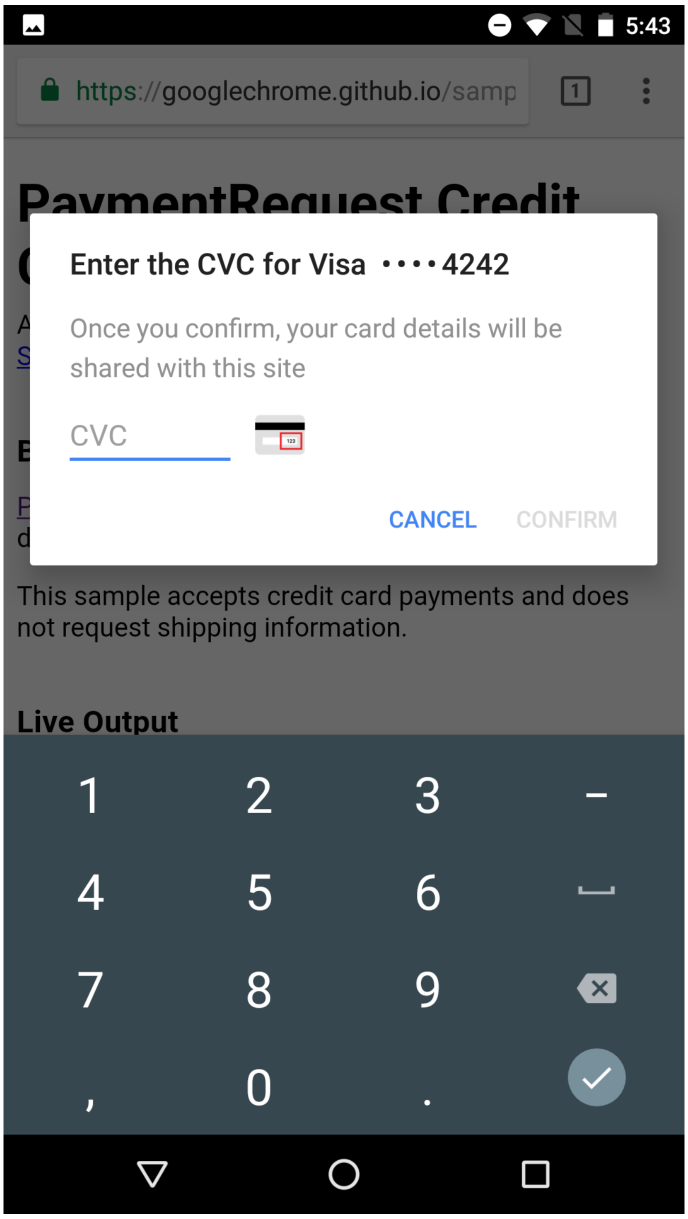
You can manage cards and addresses in Settings.



3. Process the payment

Upon user tapping on "PAY" button, a promise will be resolved and payment information will be passed to the resolving function. You can send the information either to your own server, or send it through a third party like Stripe for processing.

```
request.show()
.then(result => {
  // POST the payment information to the server
  return fetch('/pay', {
    method: 'POST',
    credentials: 'include',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(result.toJSON())
  }).then(response => {
    // Examine server response
    if (response.status === 200) {
      // Payment successful
      return result.complete('success');
    } else {
      // Payment failure
      return result.complete('fail');
    }
  }).catch(() => {
    return result.complete('fail');
  });
});
```



5:43

https://googlechrome.github.io/samp

PaymentRequest Credit

Enter the CVC for Visa ••••4242

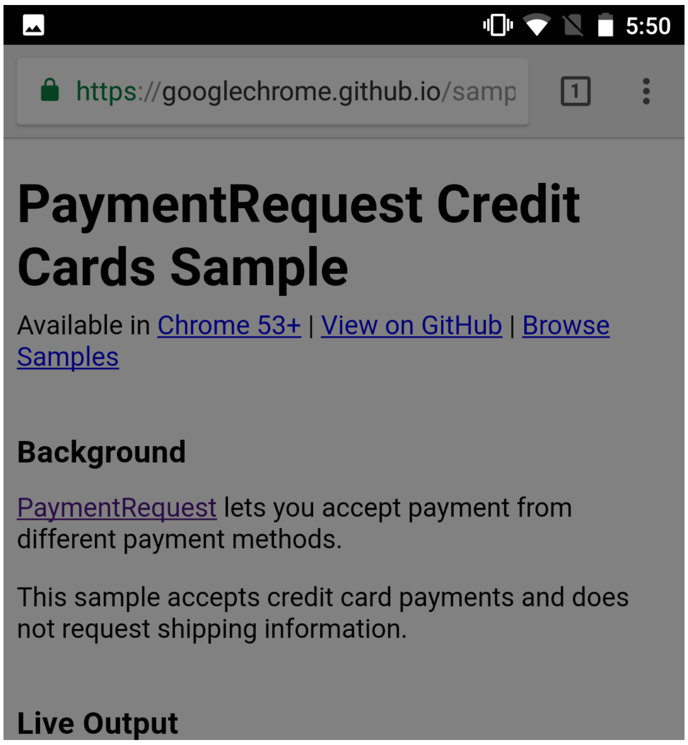
Once you confirm, your card details will be shared with this site

CVC

CANCEL CONFIRM

This sample accepts credit card payments and does not request shipping information.

Live Output



5:50

https://googlechrome.github.io/samp

PaymentRequest Credit Cards Sample

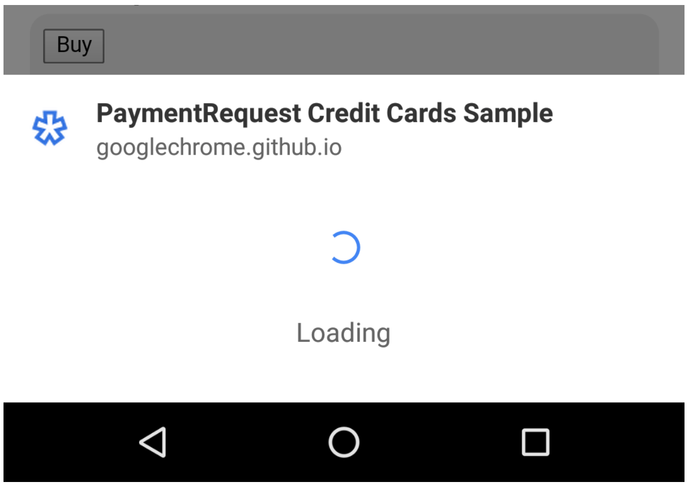
Available in Chrome 53+ | View on GitHub | Browse Samples

Background

PaymentRequest lets you accept payment from different payment methods.

This sample accepts credit card payments and does not request shipping information.

Live Output



4. Display payment results

If the payment verification was successful, call `.complete('success')` to complete the purchase, otherwise `.complete('fail')`. Success / failure status will be displayed using a native UI. Upon resolving the `.complete()`, you can proceed to the next step.

Payment Request API can do more

Shipping items

If you are selling physical goods, you'll probably need to collect the user's shipping address and a shipping preference such as "Free shipping" or "Express shipping". Payment Request API certainly supports those use cases. See the integration guide to learn more.

Adding more payment solutions

Credit card is not the only supported payment solution for Payment Request. There are a number of other payment services and solutions in the wild and the Payment Request API is designed to support as many of those as possible. Google is working to bring Android Pay to Chrome. Other third party solutions will be supported in the near future as well. Stay tuned for updates.

FAQ

Are there any restrictions to using the API?

Use Chrome for Android with version 53 or later. Requires secure origin - HTTPS, localhost or file:///.

Is it possible to query the available payment methods?

Currently not supported, but we're investigating ways of exposing this API in a privacy-sensitive way. Payment Request API is designed to support the broadest possible array of payment methods. Each payment method is identified by a payment method identifier. Payment Method Identifiers will support distributed extensibility, meaning that there does not need to be a central machine-readable registry to discover or register payment methods.

Do you plan on offering a coupon code?

We are investigating how to best do this. For now, you can manually ask for a coupon code before or after calling the API.

Does this work with iframes?

Starting with Chrome 57, using the Payment Request API inside an **`iframe`** is supported. Simply add **`allowpaymentrequest`** attribute to allow the **`iframe`** to use the API.

```
<iframe src="URL_INCLUDING_PAYMENT_REQUEST_CALL" allowpaymentrequest></ifra
```

Are there any polyfills available to support incompatible browsers for this API?

Not currently.

Can I assume the current API is final?

It could change. We provide a shim that protects you from API changes that may be backwards incompatible. By embedding the shim in your website, it will paper over any API differences for two major Chrome versions.

Resources

To learn more about Payment Request API, a few documents and resources are available:

- Official specification
- Payment Request API integration guide
- Demo
- Simple demos and sample code

Last updated July 2, 2018.