Creating a Proof of Concept for Payment Request API

Ian Jacobs
<ij@w3.org>
Goals for a POC

• Demonstrate a streamlined user experience for a Web payment

• Evaluate the cost and complexity of developing this experience with new Web standards
Disclaimer

- I am not endorsing any particular browser or product.
- This deck makes some recommendations for prototyping based on implementation status as of the date of this presentation; recommendations might change over time.
Benefits of W3C Payment Request

• Improved user experience in a variety of ways including: faster checkout, consistency across the Web, and modal experience keeping customer near merchant context.
• Easier development of checkout pages for merchants and their service providers.
• Improved security by making it easier to bring innovative payment methods to the Web.
• More customer-facing interactions for payment handler distributors.
General Flow Diagram

1. **Payee (may be a PSP)**
   - Provide checkout page

2. **Browser/User Agent**
   - Click buy button

3. **Payer**
   - Call Payment Request API
   - Show matching payment handlers
   - Select a payment handler
   - Launch selected payment handler
   - User auth and other interaction
   - Response data after user interaction

4. **Payment Handler**
   - Payment Request API Response

5. **User PSP**
   - Handler communication with server
What’s Involved in a POC

- Payment method

- Merchant web site that calls Payment Request API and accepts that payment method

- Payment handler that implements the payment method and enables the user to make a payment

- Browser, which mediates the request from the merchant and response by the user.
Connecting the Elements

- **Merchant Web Site** (or gateway)
- **Browser**
- **Web-based Payment Handler**
- **Server of Payment Method Owner**
  - Hosts payment method manifest
  - Hosts Web app manifest
- **Payment server** (e.g., bank, token service provider, etc.)

Payment Request API

Payment Handler API
Stored Data the Payee can Request

• Managed by the mediator (browser)
  • Payer name
  • Payer email
  • Payer phone
  • Shipping address
  • Preferred shipping option (shipping, delivery, pickup)
  • Billing address

• Managed by the payment handler (whether Web-based, native, browser)
  • Payment method-specific data (may also include billing address)
The Elements
Browser

- As of 1 March 2019, use Chrome for a proof of concept.

- Other browsers support Payment Request, but today only Chrome supports an open ecosystem of third party payment handlers.
Payment Method

• A URL to identify the payment method uniquely (e.g., https://apple.com/apple-pay or https://google.com/pay)

• A JSON data model:
  - Request data provided by the merchant
  - Response data returned by the payment handler

• A Payment Method Manifest hosted on a server with the same origin as the payment method URL. This is how the payment method owner tells the browser which payment handlers are authorized. (BobPay Example)
Sample Payment Response
Data for Basic Card

dictionary BasicCardErrors {
  DOMString cardNumber;
  DOMString cardholderName;
  DOMString cardSecurityCode;
  DOMString expiryMonth;
  DOMString expiryYear;
  AddressErrors billingAddress;
};

For more information see the Basic Card Specification
Merchant Site

- The merchant site calls Payment Request API.

- Some code examples:
  - Rouslan Solomakhin examples
  - Paymentrequest.show
  - Documentation on Mozilla developer network

- See also developer guides from various vendors
Payment Handler

- A payment handler can either be native (e.g., Android) or Web-based. Recommend Web-based for the POC.

- Web-based payment handlers communicate with the browser via the Payment Handler API. (BobPay payment handler example)

- A Web-based payment handler is a Web app that leverages service workers. It also leverages the Web app manifest (BobPay example).
Considerations

• Is strong authentication required? When does it occur in the flow? Can the POC leverage Web Authentication?

• Does the merchant need to provide any request data to the payment handler (e.g., account number, merchant identification)?

• Does the payment handler talk to a server? Does that server also need to talk to the merchant (e.g., for merchant validation)?
Summary of Specs

- Core
  - Payment Request API
  - Payment Method Identifiers
  - Payment Handler API
  - Payment Method Manifest

- Payment methods
  - Basic Card
  - Drafts for: Tokenized card, credit transfer, …
More Resources

- FAQ
- Introductions to Payment Request (includes some videos)
- Developer guides
- Documentation Mozilla Developer Network