WPWG @ TPAC: PaymentHandlers + Privacy

smcgruer@google.com

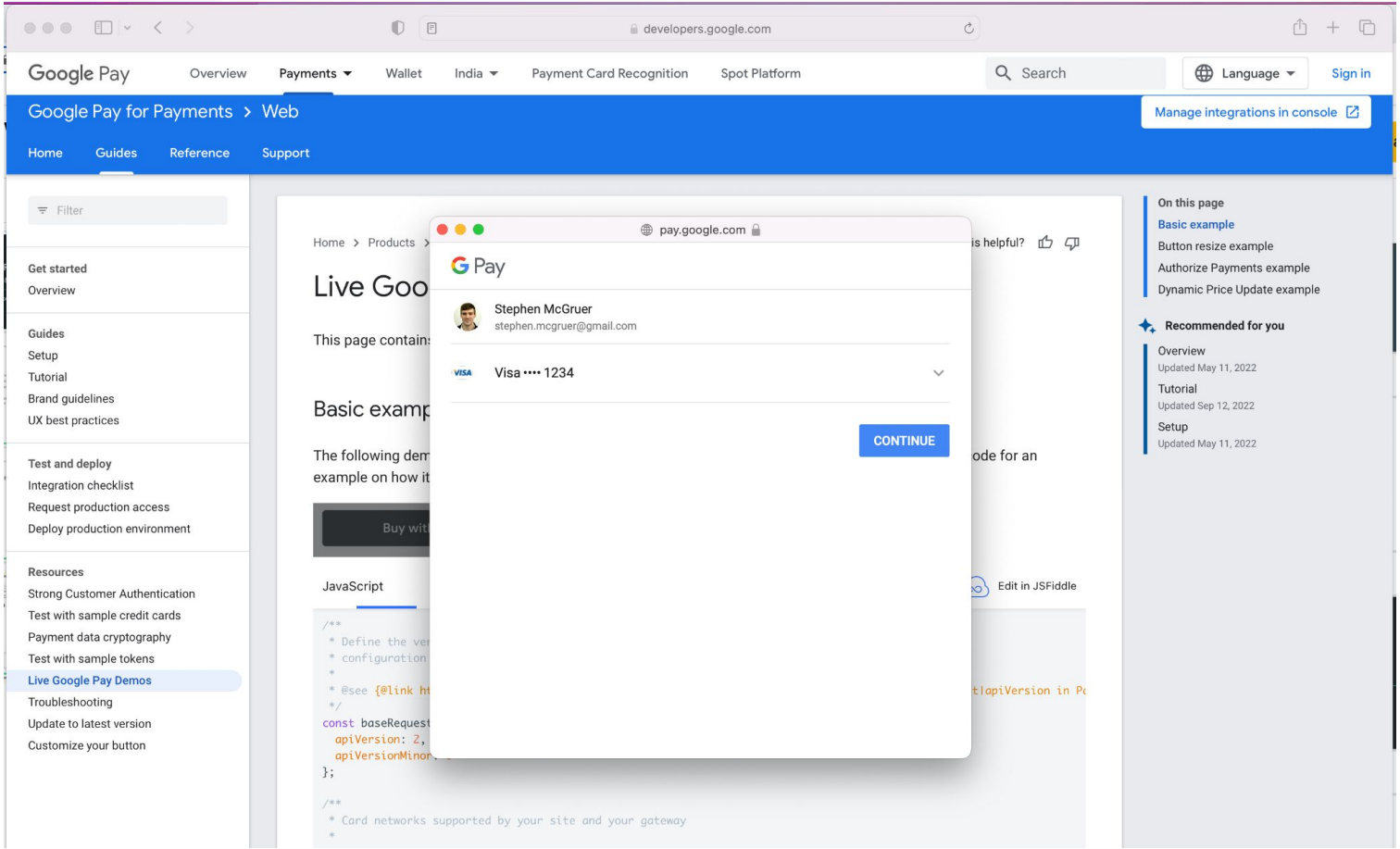
2022/09/13

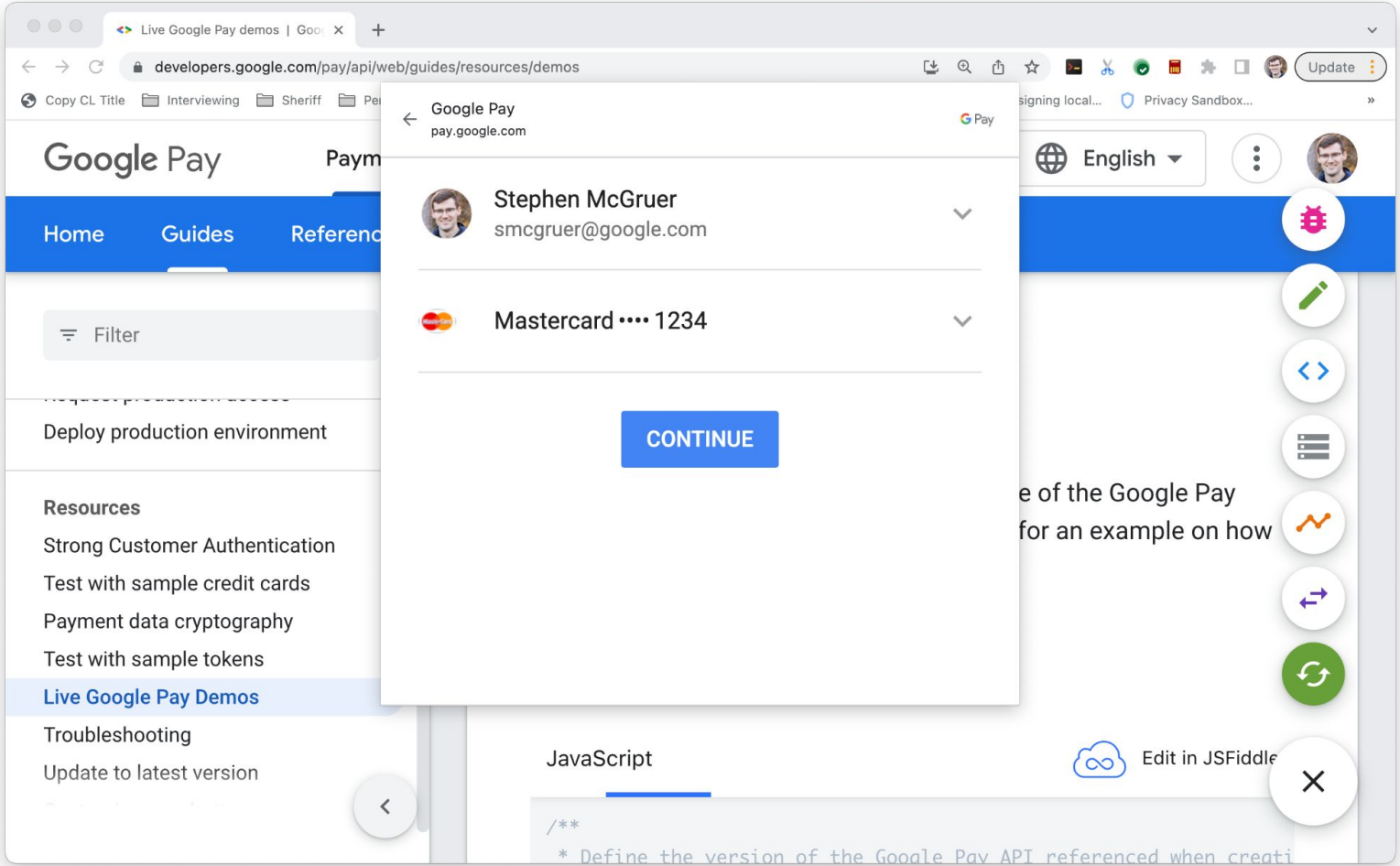
What is the Payment Handler API?

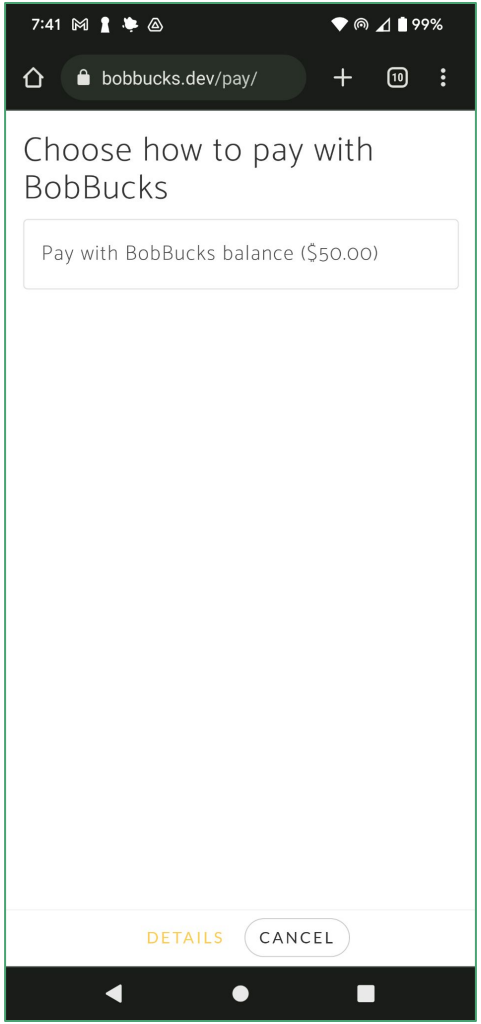
- Attempt to support an open web ecosystem of Payment Apps/Wallets
- Provides mechanism to 'install' a web-based Payment App, which is then accessed via Payment Request
  - On Chrome Android, we also support integration with natively installed Payment Apps.
- Interacts with all PaymentRequest APIs:
  - show(), retry(), canMakePayment(), hasEnrolledInstrument(), ...

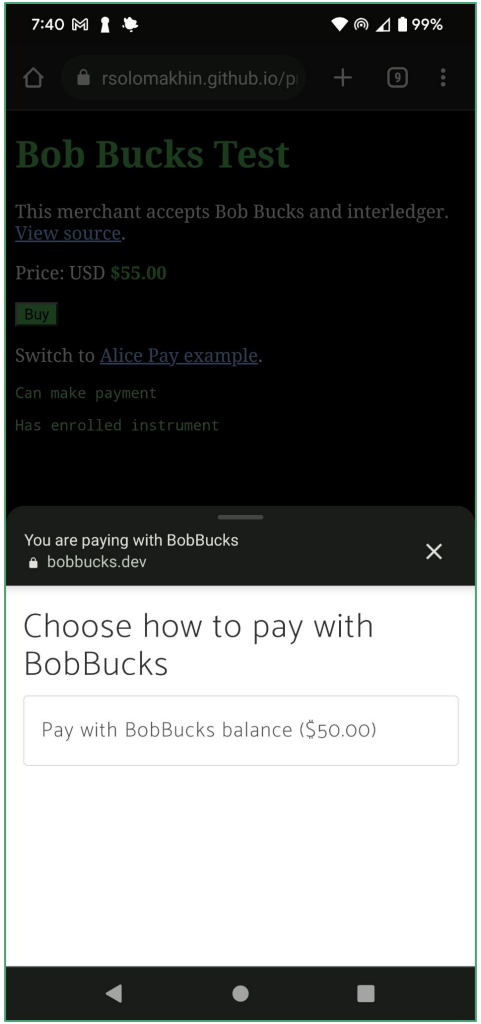
- **Note**: Not here to re-litigate their value :)

Spec


developers.google.com
Google Pay
Overview
Payments
Wallet
India
Payment Card Recognition
Spot Platform
Search
Language
Sign in
Google Pay for Payments > Web
Manage integrations in console
Home
Guides
Reference
Support
Filter
Get started
Overview
Guides
Setup
Tutorial
Brand guidelines
UX best practices
Test and deploy
Integration checklist
Request production access
Deploy production environment
Resources
Strong Customer Authentication
Test with sample credit cards
Payment data cryptography
Test with sample tokens
Live Google Pay Demos
Troubleshooting
Update to latest version
Customize your button
Home > Products >
Live Goo
This page contains
Basic examp
The following dem
example on how it
Buy wit
JavaScript
Edit in JSFiddle
/**
* Define the ver
* configuration
*
* @see {@link ht
*/
const baseRequest
apiVersion: 2,
apiVersionMinor
};
/**
* Card networks supported by your site and your gateway
*
pay.google.com
G Pay
Stephen McGruer
stephen.mcgruer@gmail.com
Visa •••• 1234
CONTINUE
s helpful?
ode for an
tlapiVersion in Pa
On this page
Basic example
Button resize example
Authorize Payments example
Dynamic Price Update example
Recommended for you
Overview
Updated May 11, 2022
Tutorial
Updated Sep 12, 2022
Setup
Updated May 11, 2022


Live Google Pay demos | Goo
developers.google.com/pay/api/web/guides/resources/demos
Update
Copy CL Title
Interviewing
Sheriff
Privacy Sandbox...
Google Pay
Paym
English
Home
Guides
Filter
Deploy production environment
Resources
Strong Customer Authentication
Test with sample credit cards
Payment data cryptography
Test with sample tokens
Live Google Pay Demos
Troubleshooting
Update to latest version
Google Pay
pay.google.com
Stephen McGruer
smcgruer@google.com
Mastercard •••• 1234
CONTINUE
e of the Google Pay
for an example on how
JavaScript
Edit in JSFiddle
/**
* Define the version of the Google Pay API referenced when creati


7:41
99%
bobbucks.dev/pay/
Choose how to pay with BobBucks
Pay with BobBucks balance ($50.00)
DETAILS
CANCEL


7:40
99%
rsolomakhin.github.io/p
Bob Bucks Test
This merchant accepts Bob Bucks and interledger.
View source.
Price: USD $55.00
Buy
Switch to Alice Pay example.
Can make payment
Has enrolled instrument
You are paying with BobBucks
bobbucks.dev
Choose how to pay with BobBucks
Pay with BobBucks balance ($50.00)

How does Payment Handler work?

- Installation - two methods:
  a. In a first-party context, install a service-worker, then call `PaymentManager.paymentInstruments.set` to make it a Payment App.
  b. Utilize the Payment Method Manifest Spec to install a Payment App (service-worker) Just In Time (JIT) during a Payment Request flow.

- In practice, nobody uses (a). This is good, because (spoiler) it's Bad News™.

- Important for later: the Payment Handler service-worker has unpartitioned storage, because it's meant to be akin to a pop-up window.

How does Payment Handler work?

- At Payment Request construction time.

```
const request = new PaymentRequest([
  {supportedMethods:
     'https://pay.example/app',
   ...}],
...);
```

- If matching service-worker already exists:
  - Fire a "canmakepayment" event at the installed service-worker
  - Record its answer for later use.
- Else:
  - Fetch https://pay.example.app, read 'payment-method-manifest' URL from Link header
  - Fetch payment method manifest URL, and parse it (it's a web app manifest, example)
  - From the manifest, fetch the 'default_applications' manifest and parse it.
  - This is the Payment App manifest (example).
  - **Nothing is installed yet**.

How does Payment Handler work?

- When canMakePayment() is called.

```
const canMakePayment =
    await request.canMakePayment();
```

- If matching service-worker already exists:
  - return true
- Else:
  - return true **if** a manifest exists that *could* install a valid service worker, or false otherwise

- Note: the result of "canmakepayment" isn't used here. It's actually for hasEnrolledInstrument. 🤦

How does Payment Handler work?

- When hasEnrolledInstrument() is called.

```
const hasEnrolledInstrument = await
    request.hasEnrolledInstrument();
```

- If matching service-worker already exists:
  - return the result of the earlier "canmakepayment" event from the service-worker
- Else:
  - return false

- Note: there is an existing 'defense' against abuse of this API in the form of a timer, but it is likely ineffective against abuse.

How does Payment Handler work?

```
const result = await request.show();
```

- When show() is called.

- If service-worker not currently installed:
  - Install service worker from the manifest.

- Fire a "paymentrequest" event at the installed service-worker.
- The service worker **may** call openWindow(url) to open the modal window.
- The service worker **may** call changePaymentMethod(...) to interact with the calling website (the merchant).
- The service worker **must** eventually call responseWith(...) with the results of the user's payment flow to the merchant.

Privacy Sandbox

- Collection of **many** different privacy initiatives from Google
- Quite complex tbh, so I will be simplifying (and also probably getting some things wrong).
- But the main concept for **this** conversation is about anti-tracking:
  - Two origins (a.com and b.com) should not be able to share their concept of who the current user is, without user knowledge / consent.
  - (Exact line between knowledge / consent is blurry, and we are still exploring this ourselves.)
- One step we're (slowly!) taking against this is third-party cookie removal, but it goes beyond that.

https://privacysandbox.com/

Privacy Sandbox + Payment Handlers - our concerns

- Main cases:
  - "canmakepayment" event carries fields that enable silent merchant.com <-> paymentapp.com communication (Plan: remove these fields - but does this make "canmakepayment" useless?)
  - PaymentInstruments.set()/get() are incredibly dangerous - completely silent sharing of info in a third-party context (Plan: remove entirely, only allow JIT-install of PaymentHandlers)
  - Payment Handlers can opt not to display any UI when show() is called - but show() has opened a communication channel between the two first-party contexts! (Plan: force always showing UI, which informs the user that a communication channel has opened. Similar to pop-up windows).
  - More complex timing attacks or attacks via installing many Payment Apps (Plan a little more TBD).

GitHub issue/pull request

But More Generally...

- I want to talk about what a privacy-preserving payment app ecosystem 'should' look like, if there's appetite.
  - What needs do payment apps have from the web, and are they (and will they) be served via existing API surfaces?
  - Is there more value we could unlock for Payment Apps with different, deeper, ??? integration, whilst **also** improving the user's control over their privacy?
  - Can we ever do 'isReadyToPay' style APIs in a privacy-preserving way? (E.g., hasEnrolledInstrument, but also non-PaymentRequest APIs that are 3p cookie or bounce based today)
- Perhaps in a breakout session tomorrow?

https://privacysandbox.com/