



PAYMENT REQUEST API



GUILHERME RV COELHO

@GRVCOELHO



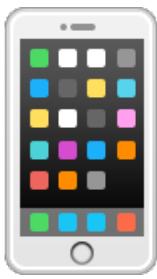


**BUYING ONLINE SHOULD BE EASY AND
SIMPLE**



**>50% OF MOBILE PURCHASES
ARE MADE THROUGH *wesites*¹**

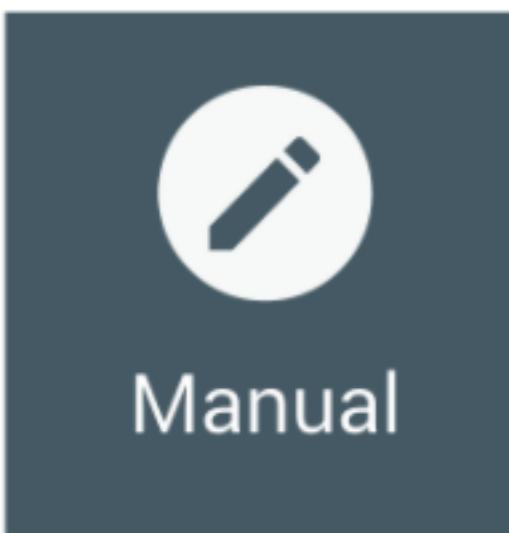
¹ Source: The State of Mobile Commerce 2016 - CRITEO 2016



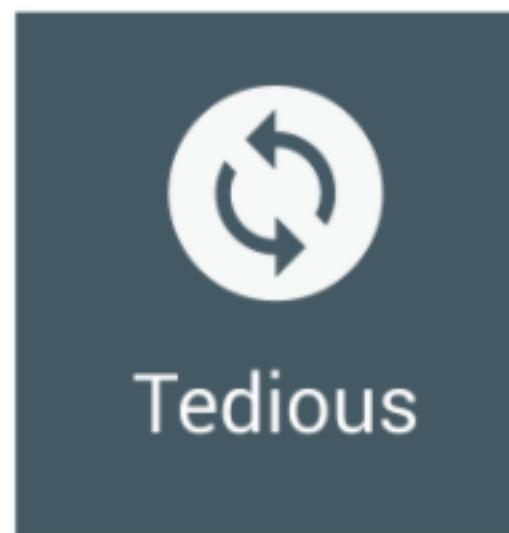
**APPS CONVERT
3X MORE THAN websites¹**

¹ Source: The State of Mobile Commerce 2016 - CRITEO 2016

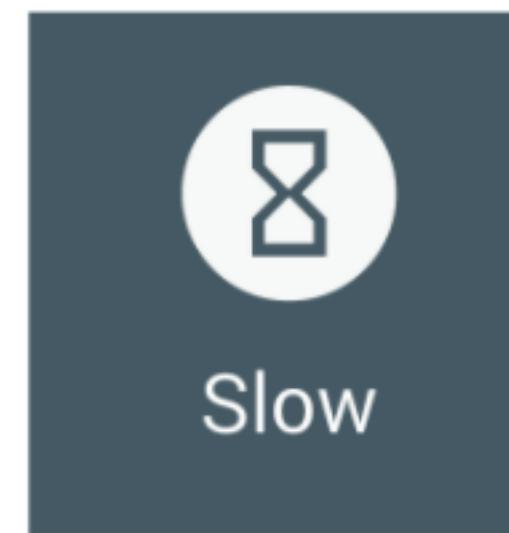
WHY?



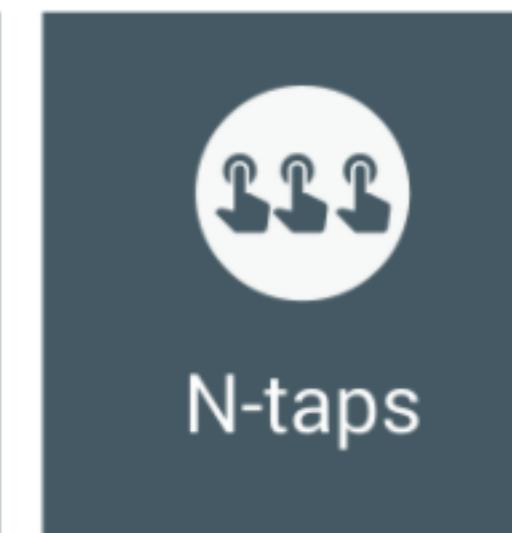
Manual



Tedious



Slow



N-taps



Your Cart / Payment / Confirmation

+ show cart

Your Information

 First Name Last Name Email

Billing Address

 Address 1 Address 2 City ST Zip

copy ↗

 Address 1 Address 2 City ST Zip

Credit Card Information

 Credit Card Number Expiration Date CVV

Contact Composition

7180 West Alaska Drive
Belmar Shopping District
Lakewood, Colorado

✉ 303.894.0025
✉ info@shopcomposition.com

Corporate Account?

Contact us to set up a corporate account.

Policies

🔗 Shopping Policies
🔗 Legal Policy

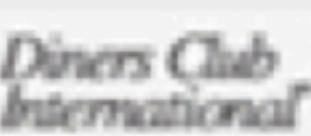
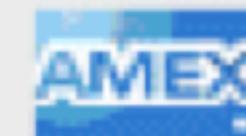
Secure Payment



🔒 Transactions 256bit encrypted.

STEP 2 OF 3: PAYMENT DETAILS

To verify your card and billing address we'll charge £2 to your card and then refund it straight away. This important check is to protect you and to make sure your order goes through smoothly.

[Learn more](#)[Learn more](#)

Type of card*

Card number*

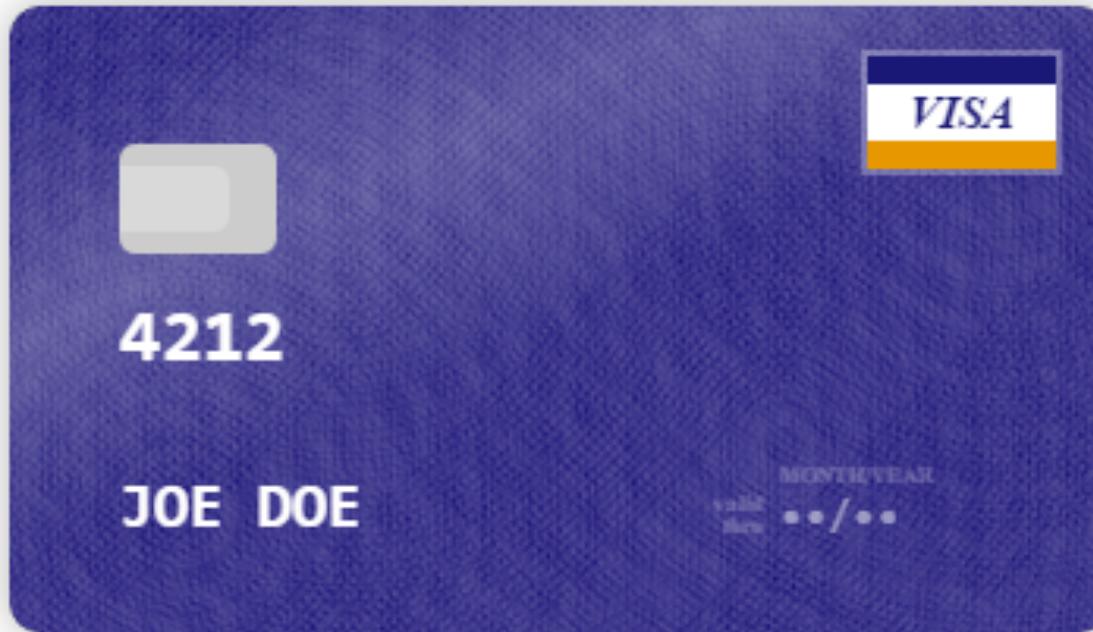
Name on card*

Security code*

Expiry date*

 /

Credit Card (saved)



Name on Card *

Joe Doe

* Required Fields

Credit Card Type *

Visa

Credit Card Number *

4212

Expiration Date *

Month

Year

Card Verification Number *

75

[What is this?](#)

CONTINUE

Checkout

Billing

Shipping

Method

Review

Payment



Payment Method

Card ▾

*Card Type

- Select-
- Select-
- American Express
- Discover
- MasterCard
- Visa

*Card Number

*CCID

Expiration Date

*Month

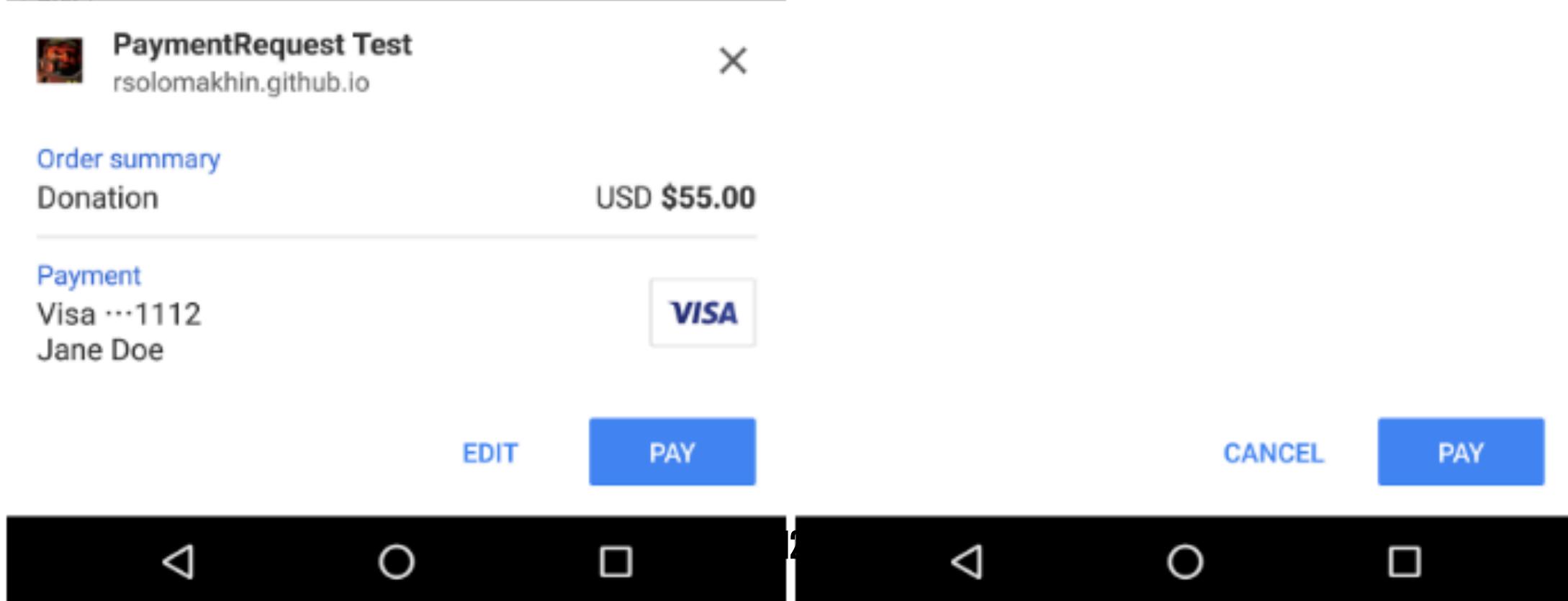
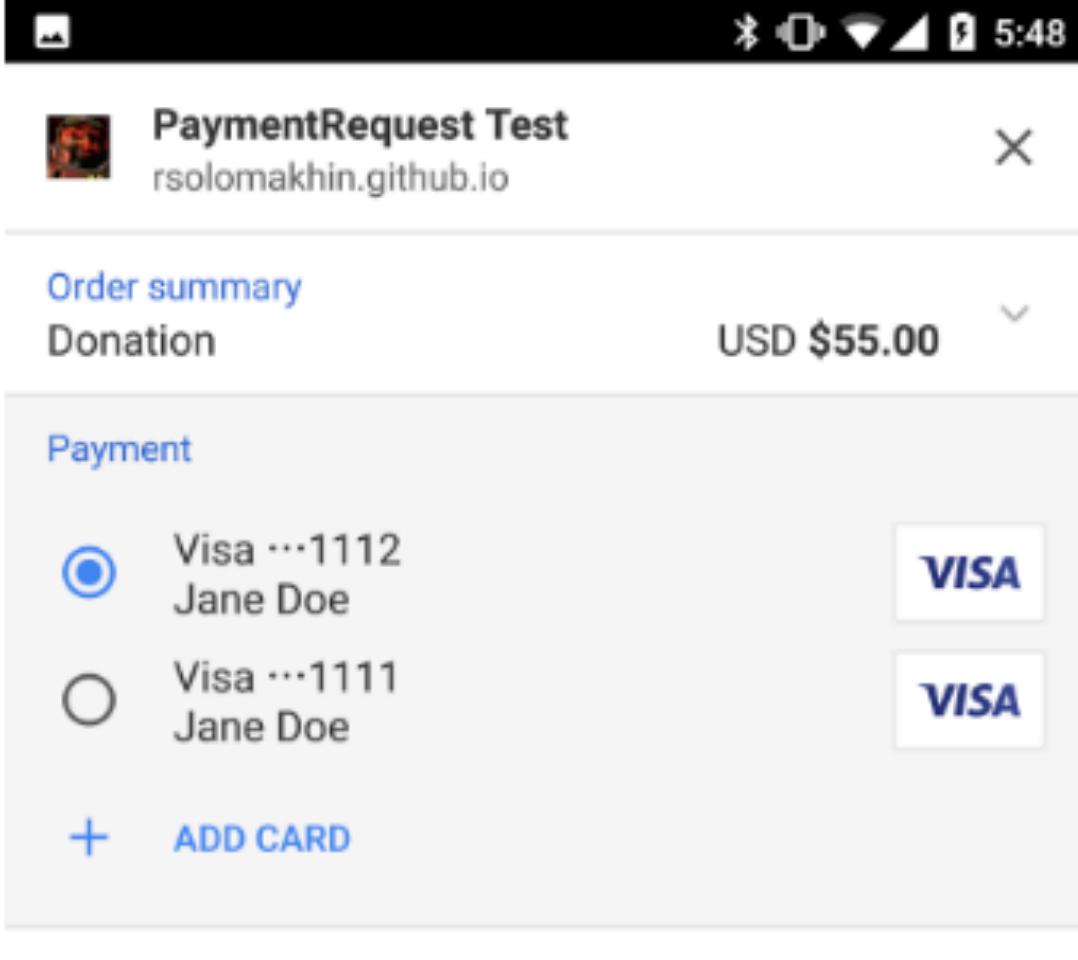
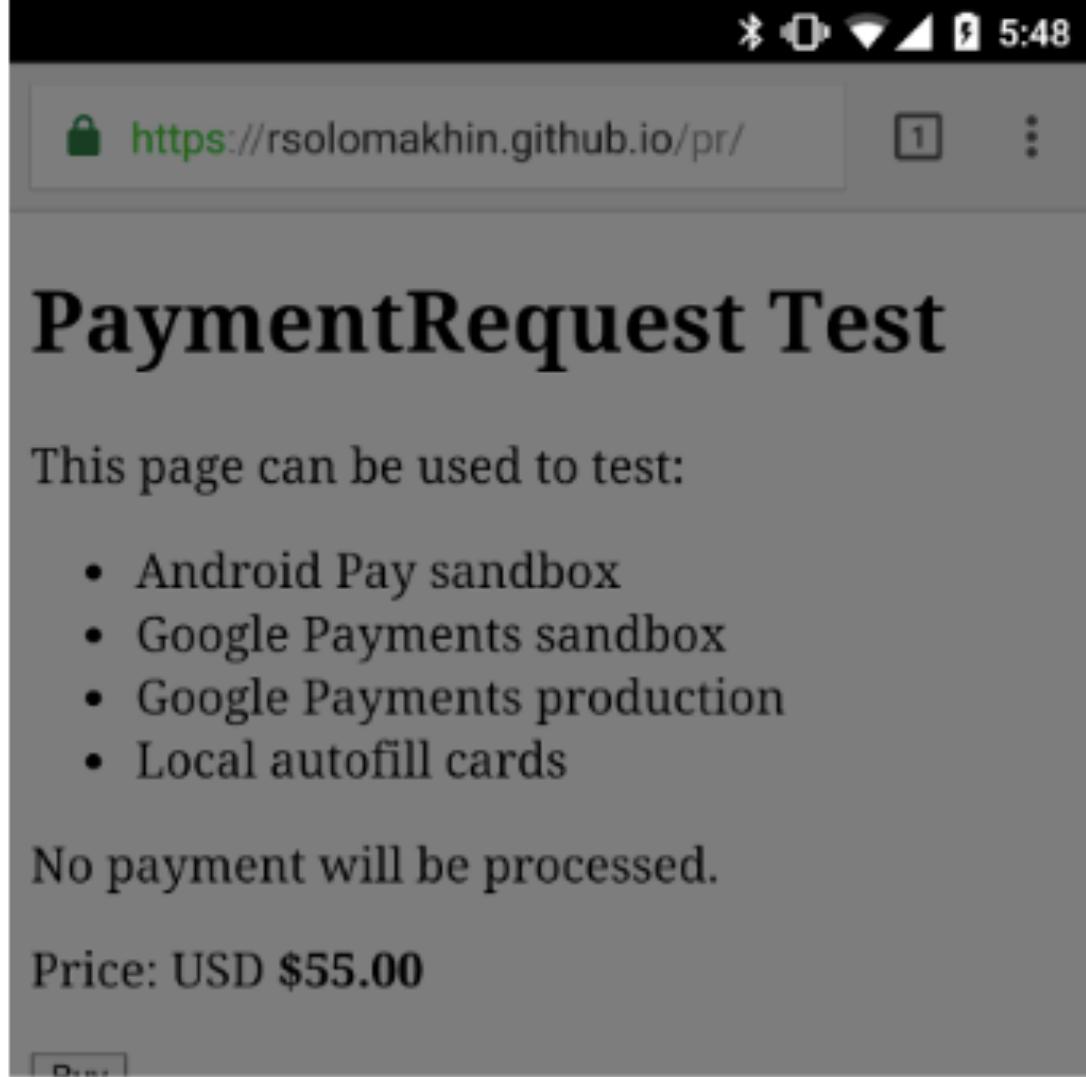
-Select- ▾

*Year

-Select- ▾



INTRODUCING PAYMENT REQUEST API





wow

The screenshot shows a web browser window displaying the W3C Editor's Draft for the Payment Request API. The title bar reads "W3C Payment Request API". The URL in the address bar is <https://w3c.github.io/browser-payment-api/>. The page content includes a sidebar titled "W3C Editor's Draft" containing a table of contents with 15 items, numbered 3 through 15. The main content area features the title "Payment Request API" and the date "W3C Editor's Draft 28 September 2016". It also includes links for "This version", "Latest published version", "Latest editor's draft", "Editors", "Version control", and copyright information. The "Abstract" section describes the API as a way for merchants to accept payments from different methods. The "Status of This Document" section notes that other documents may supersede this one.

3. Dependencies

4. PaymentRequest interface

4.1 PaymentRequest constructor

4.2 show()

4.3 abort()

4.4 State transitions

4.5 shippingAddress

4.6 shippingOption

4.7 Internal Slots

5. PaymentMethodData dictionary

6. PaymentCurrencyAmount

7. PaymentDetails dictionary

8. PaymentDetailsModifier dictionary

9. PaymentOptions dictionary

10. PaymentItem dictionary

11. PaymentAddress interface

12. PaymentShippingOption dictionary

13. PaymentResponse interface

13.1 complete()

13.2 Internal Slots

14. PaymentRequest and iframes

14.1 HTMLIFrameElement extension

15. Events

Payment Request API

W3C Editor's Draft 28 September 2016

This version: <https://w3c.github.io/browser-payment-api/>

Latest published version: <https://www.w3.org/TR/payment-request/>

Latest editor's draft: <https://w3c.github.io/browser-payment-api/>

Editors:

Adrian Bateman, Microsoft Corporation
Zach Koch, Google
Roy McElmurry, Facebook

Version control:

[Github Repository](#)
[Issues](#)

Copyright © 2016 W3C® (MIT, ERCIM, Keio, Beihang). W3C [liability](#), [trademark](#) and [permissive document license](#) rules apply.

Abstract

This specification describes a web API to allow merchants (i.e. web sites selling physical or digital goods) to easily accept payments from different payment methods with minimal integration. User agents (e.g., browsers) will facilitate the payment flow between merchant and user.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede

<https://w3c.github.io/browser-payment-api/>

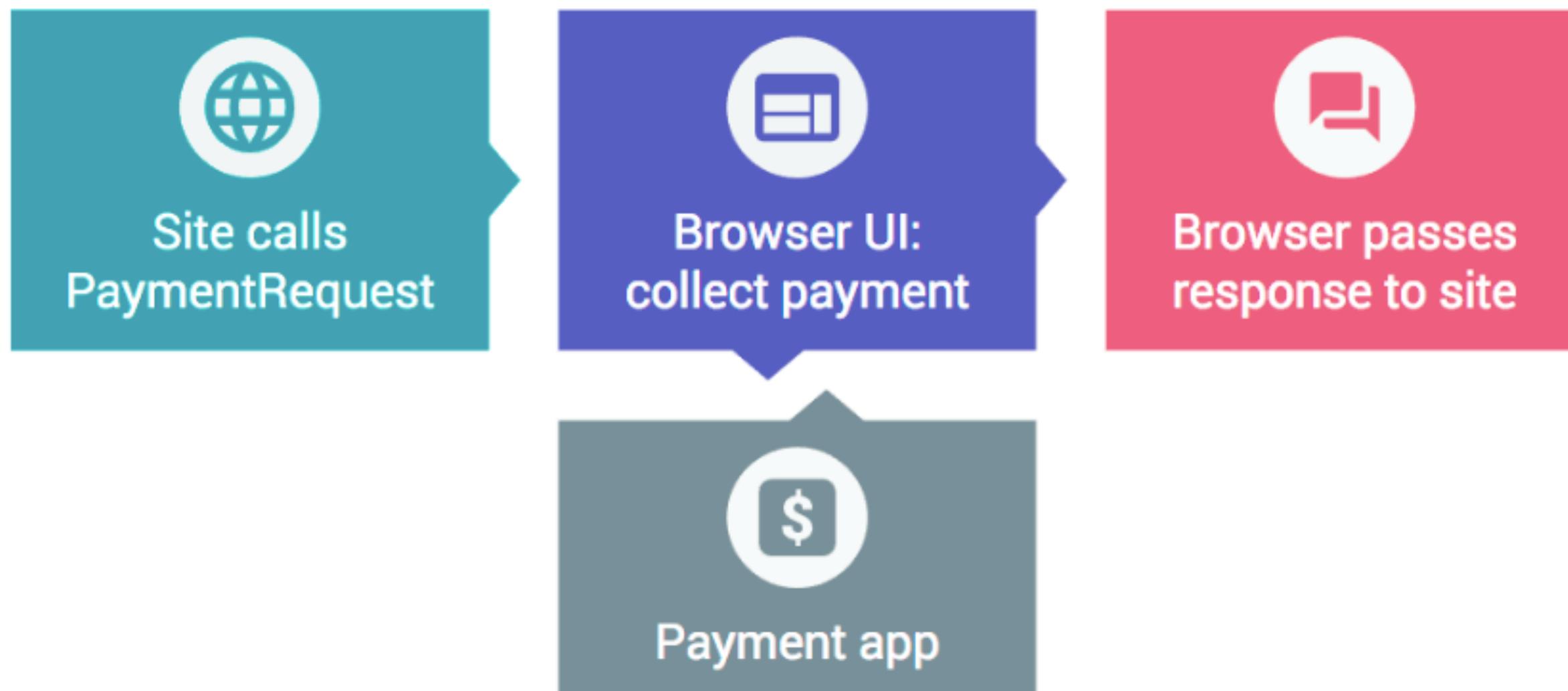
GOALS

- ▶ To let the browser be an intermediary among merchants, users, and payment methods
- ▶ To standardize the payment communication flow as much as possible
 - ▶ To seamlessly support different secure payment methods
 - ▶ To work on any browser, device, or platform—mobile or otherwise

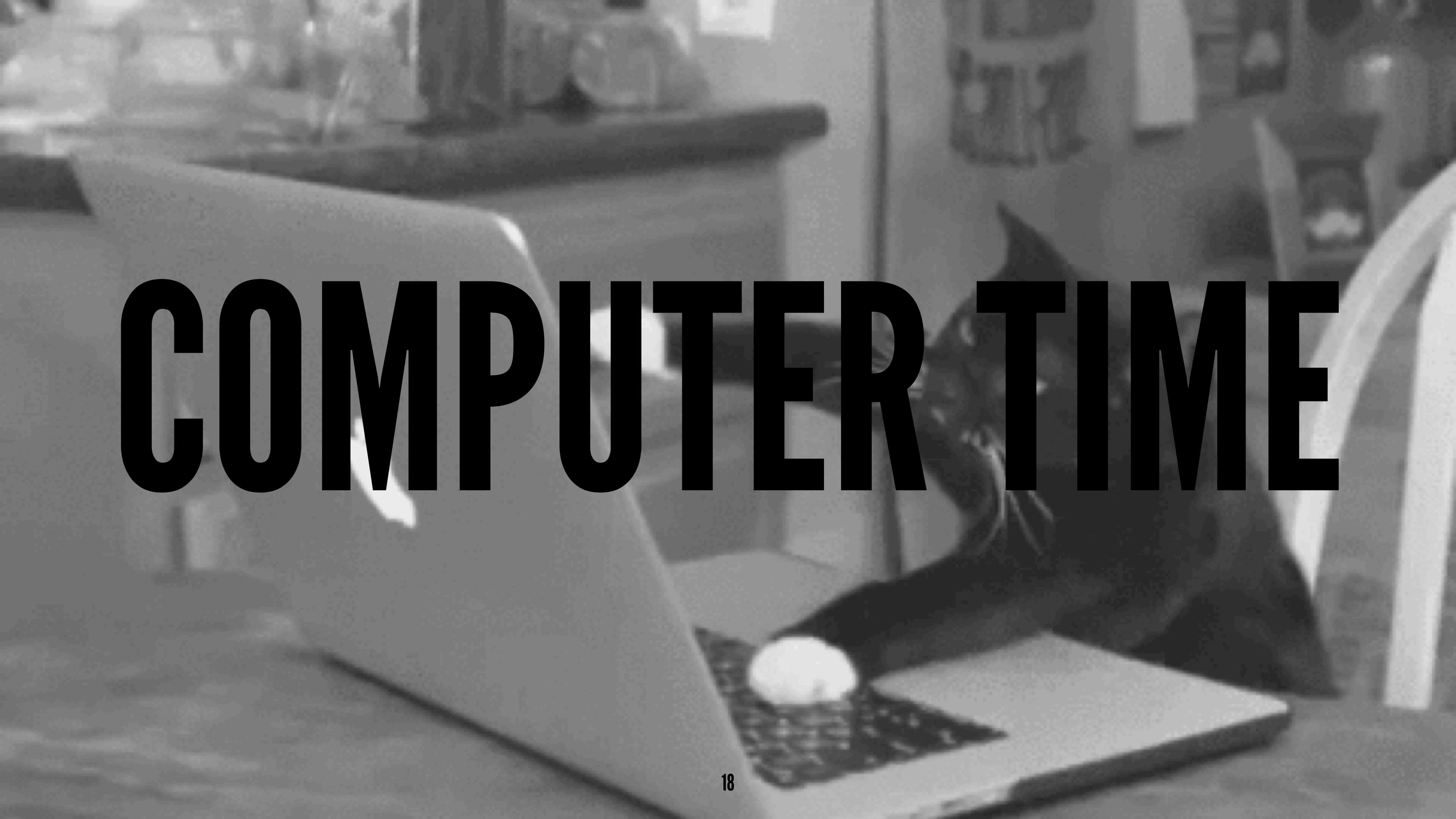
NON-GOALS

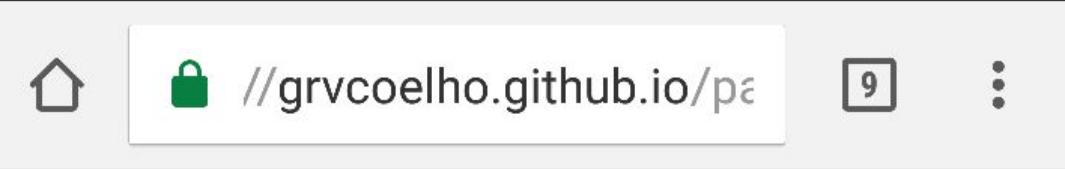
- ▶ Not trying to create a new payment method
- ▶ Not trying to integrate directly with payment processors

PAYMENT TRANSACTION PROCESS



COMPUTERTIME





Aerospacial and Thermonuclear Club



<https://grvcoelho.github.io/payment-request>

Welcome to Aerospacial and Thermonuclear Club (c) 2020

Here you can have access to rocket engines, uranium, hyperfuel and also orange juice.

Join

1. CREATE A PAYMENT REQUEST INSTANCE

```
let request = new PaymentRequest(  
    supportedInstruments,  
    details,  
    options  
)
```

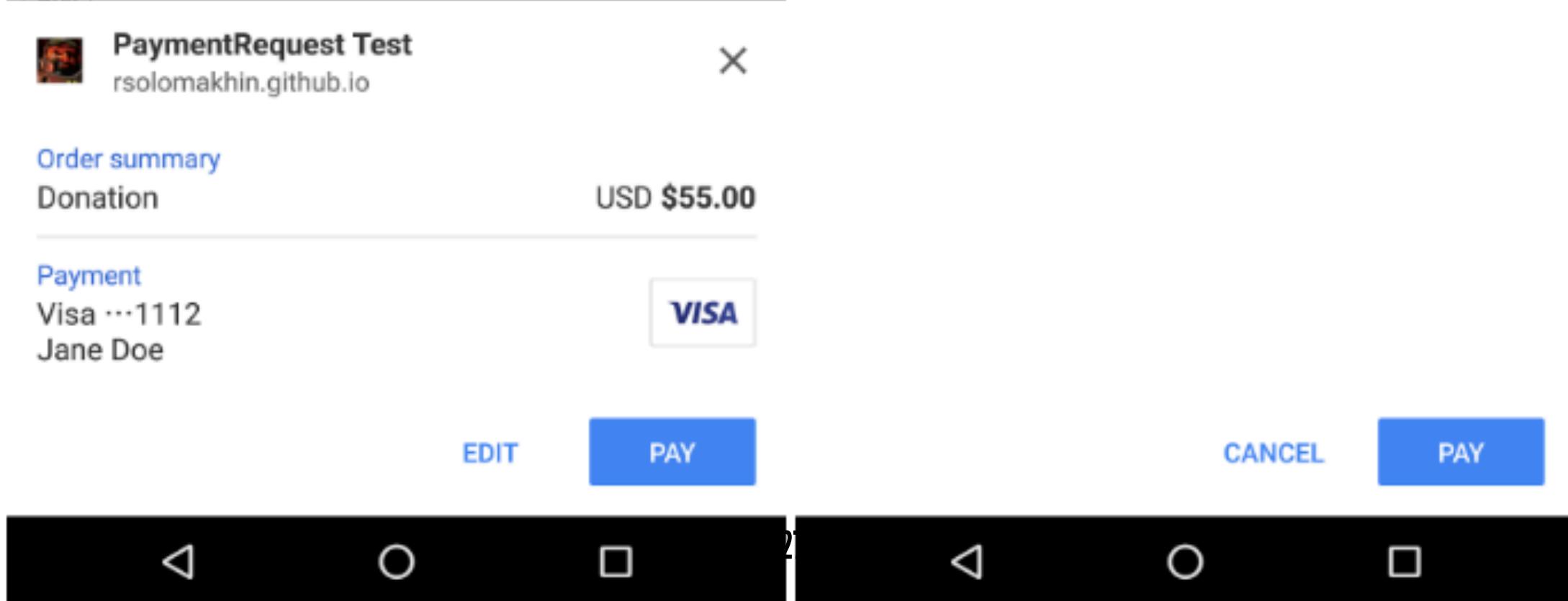
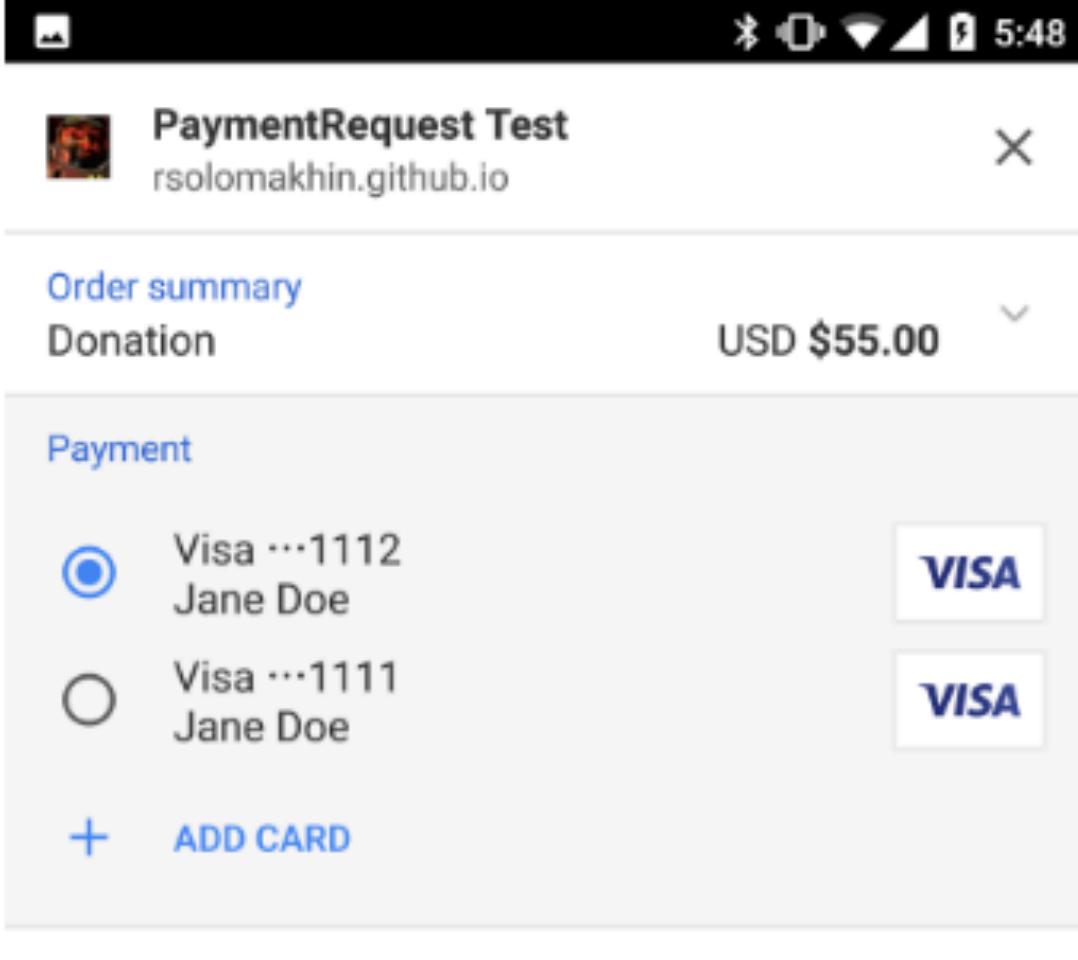
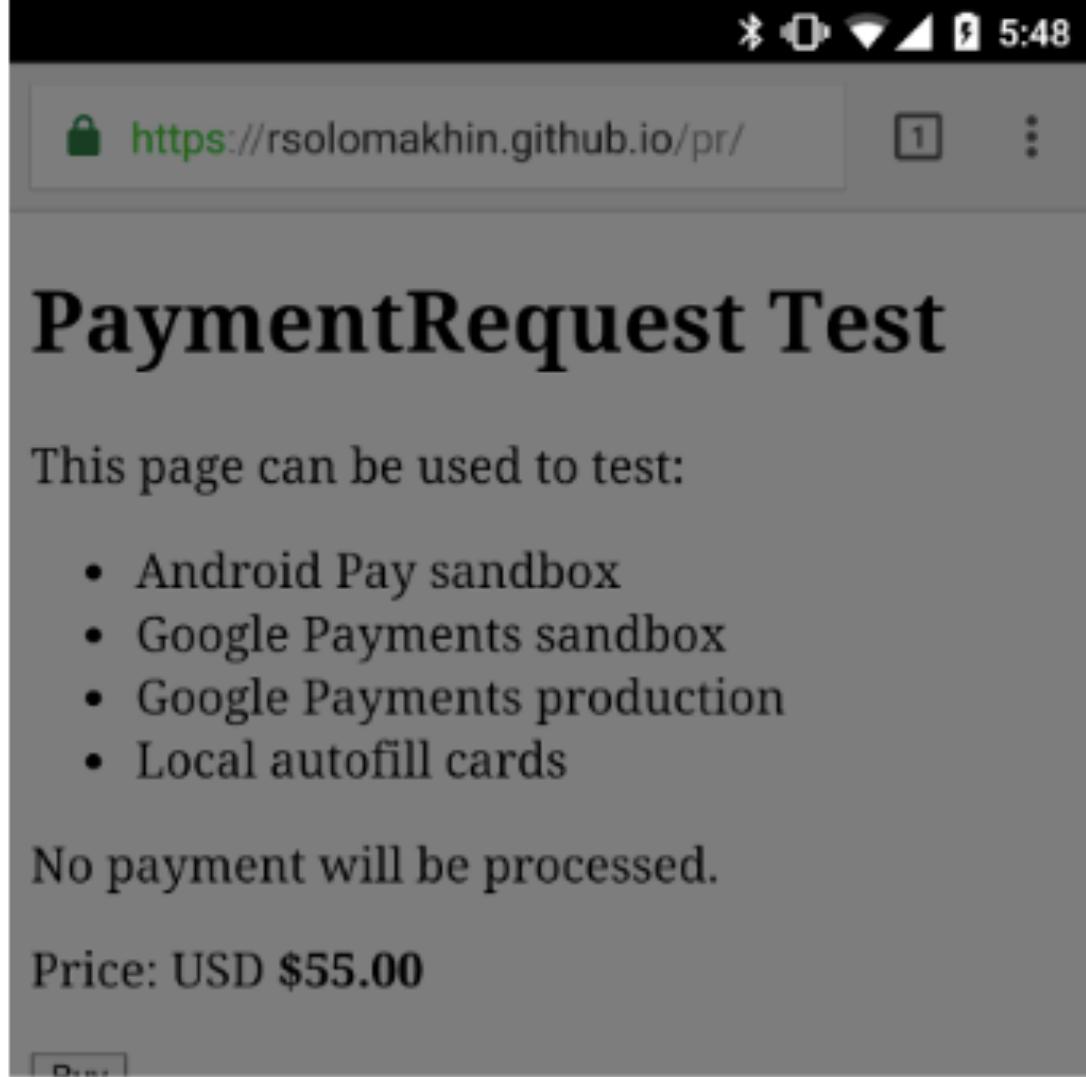
```
let supportedInstruments = [ {  
    supportedMethods: ['visa', 'mastercard']  
}]
```

```
let details = {  
    displayItems: [{  
        label: 'Original subscription amount',  
        amount: { currency: 'BRL', value : '65.00' }  
    }, {  
        label: 'Friends and family discount',  
        amount: { currency: 'BRL', value : '-10.00' }  
    }],  
  
    total: {  
        label: 'Total',  
        amount: { currency: 'BRL', value : '55.00' }  
    }  
}
```

```
let options = {  
    requestShipping: true,  
    requestPayerEmail: true,  
    requestPayerPhone: true  
}
```

**2. SHOW THE NATIVE UI
WITH .show()**

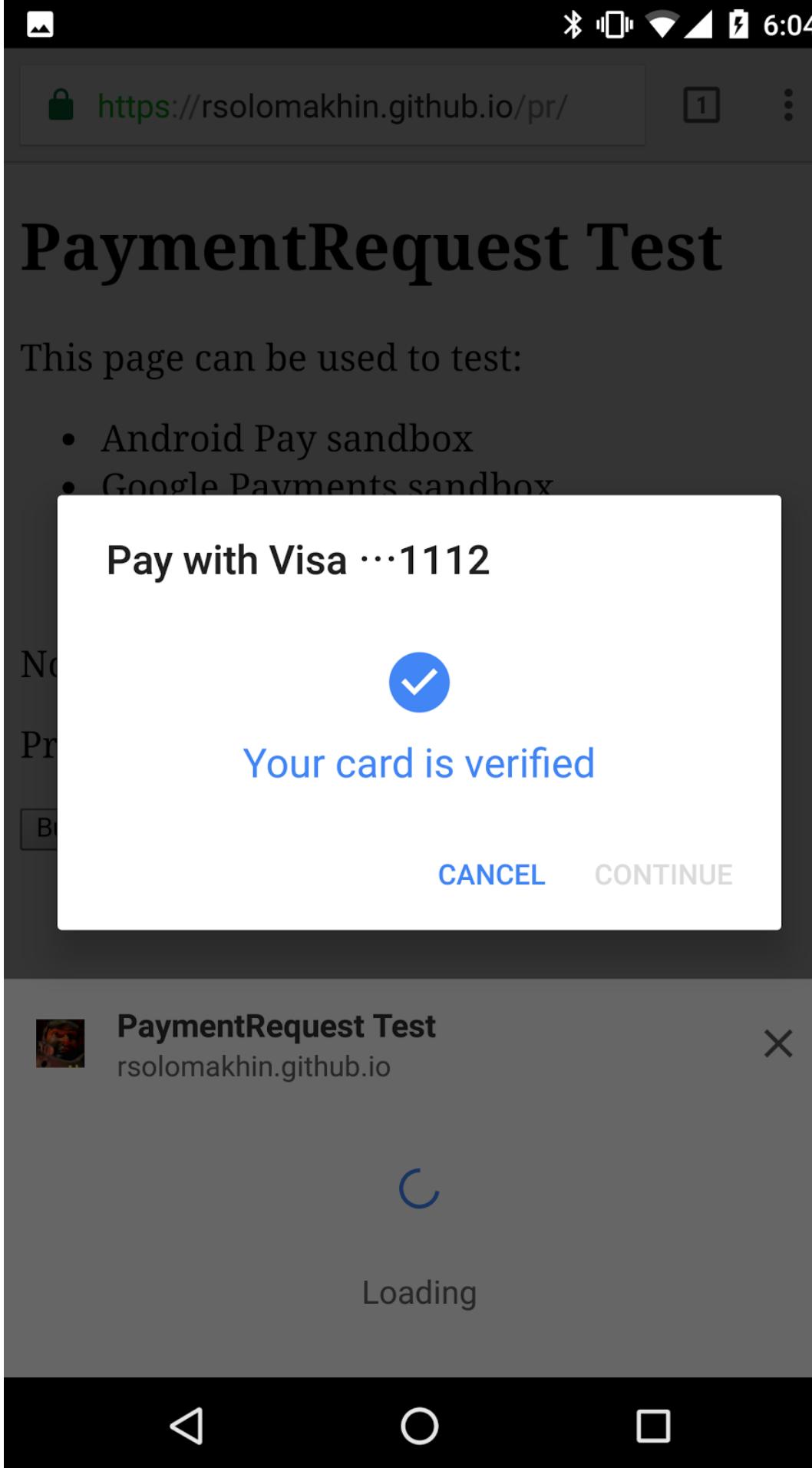
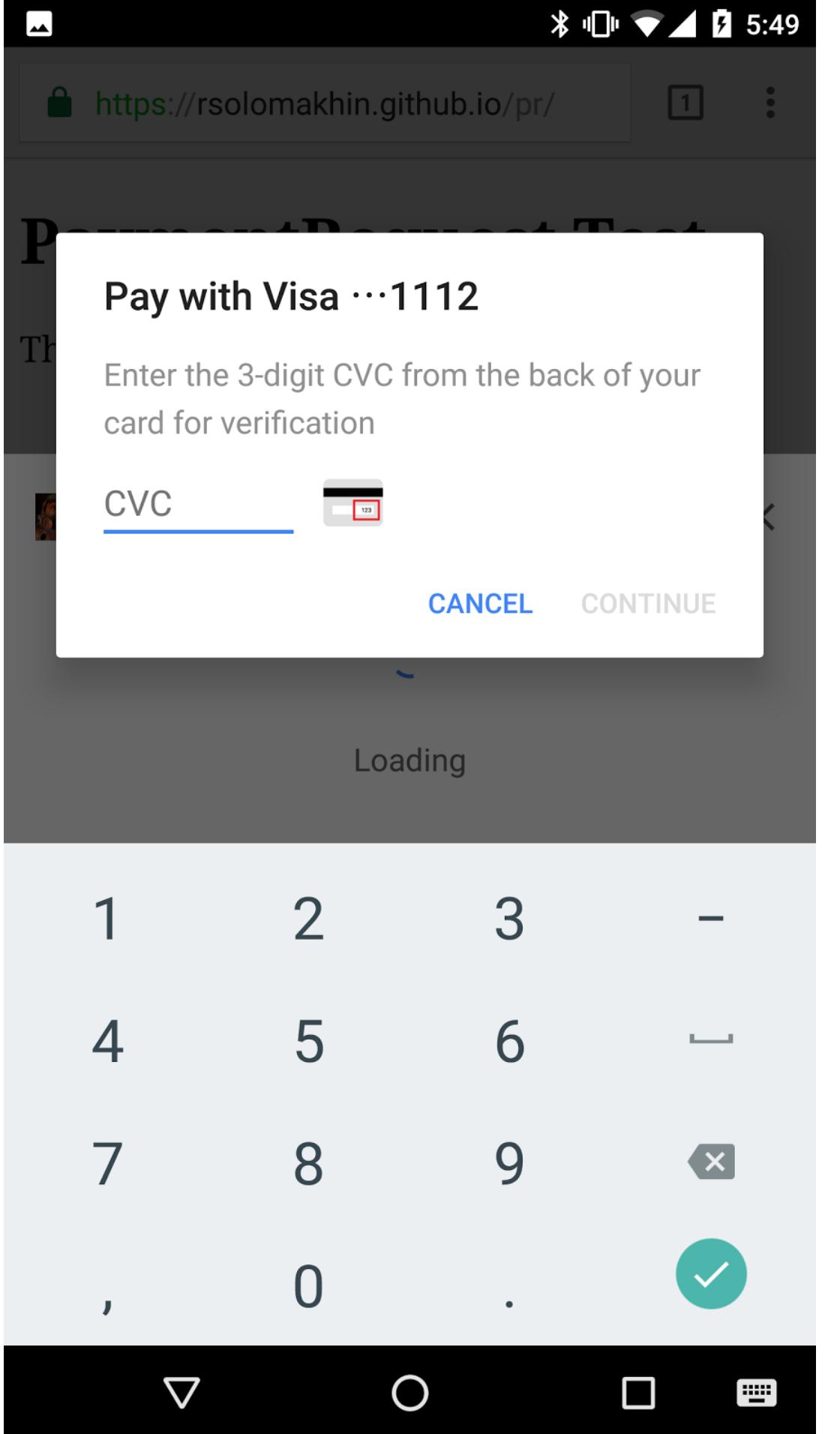
```
request.show()  
.then(payment => {  
    // User pressed "Pay"  
})
```



3. PROCESS THE PAYMENT

```
request.show().then(result => {
  let payload = {
    amount: 5500,
    encryption_key: PAGARME_ENCRYPTION_KEY,
    card_number: result.details.cardNumber,
    card_holder_name: result.details.cardholderName,
    card_cvv: result.details.cardSecurityCode,
    card_expiration_date: result.details.expiryMonth + result.details.expiryYear.substr(2, 2),
  }

  return fetch('https://api.pagar.me/1/transactions', {
    method: 'POST',
    body: JSON.stringify(payload)
  })
})
```



4. DISPLAY PAYMENT RESULTS

```
fetch('https://api.pagar.me/1/transactions', {
  method: 'POST',
  body: JSON.stringify(payload)
})
  .then(res => {
    if (res.ok) {
      result.complete('success')
    }
  })
  return result.complete('fail')
})
```

5. MAKE IT *progressive*

```
if (window.PaymentRequest) {  
    // browsers that support payment request  
} else {  
    // browsers that don't support payment request  
    pagarme.checkout.open()  
}
```



WORKS ONLY OVER HTTPS

```
$ npm install webpack-dev-server
```

```
// package.json
"scripts": {
  "start": "webpack-dev-server --https --port 3000"
}
```



WORKS ONLY ON MOBILE



**CHROME 53+
BROWSER SUPPORT**

[grvcoelho / payment-request](https://github.com/grvcoelho/payment-request)

Code Issues 0 Pull requests 1 Projects 0 Wiki Pulse Graphs Settings

Implementation of Payment Request Api using pagarme — Edit

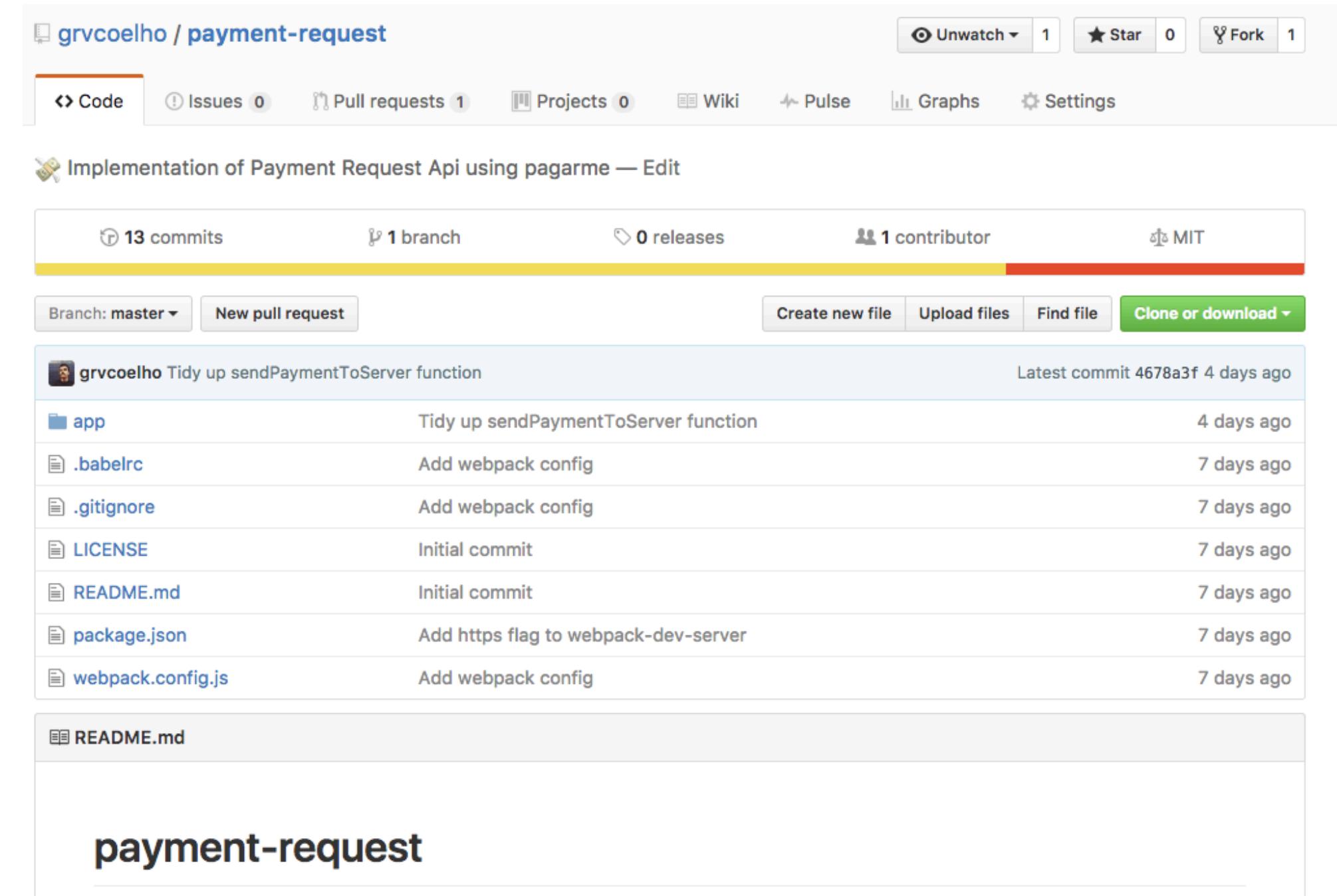
13 commits 1 branch 0 releases 1 contributor MIT

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

| grvcoelho Tidy up sendPaymentToServer function | Latest commit 4678a3f 4 days ago |
|---|----------------------------------|
| app Tidy up sendPaymentToServer function | 4 days ago |
| .babelrc Add webpack config | 7 days ago |
| .gitignore Add webpack config | 7 days ago |
| LICENSE Initial commit | 7 days ago |
| README.md Initial commit | 7 days ago |
| package.json Add https flag to webpack-dev-server | 7 days ago |
| webpack.config.js Add webpack config | 7 days ago |

README.md

payment-request



<https://github.com/grvcoelho/payment-request>



THANKS

