

#WWDC18

# Create Great Customer Experiences Using Wallet and Apple Pay

Session 720

Ben Chester, Apple Pay and Wallet Software Engineer  
Katie Calabro, Apple Pay and Wallet Software Engineer

Updates from the last year

Updates from the last year

Great experiences with Apple Pay

Updates from the last year

Great experiences with Apple Pay

Getting the most from passes

Updates from the last year

Great experiences with Apple Pay

Getting the most from passes

Digital to physical commerce

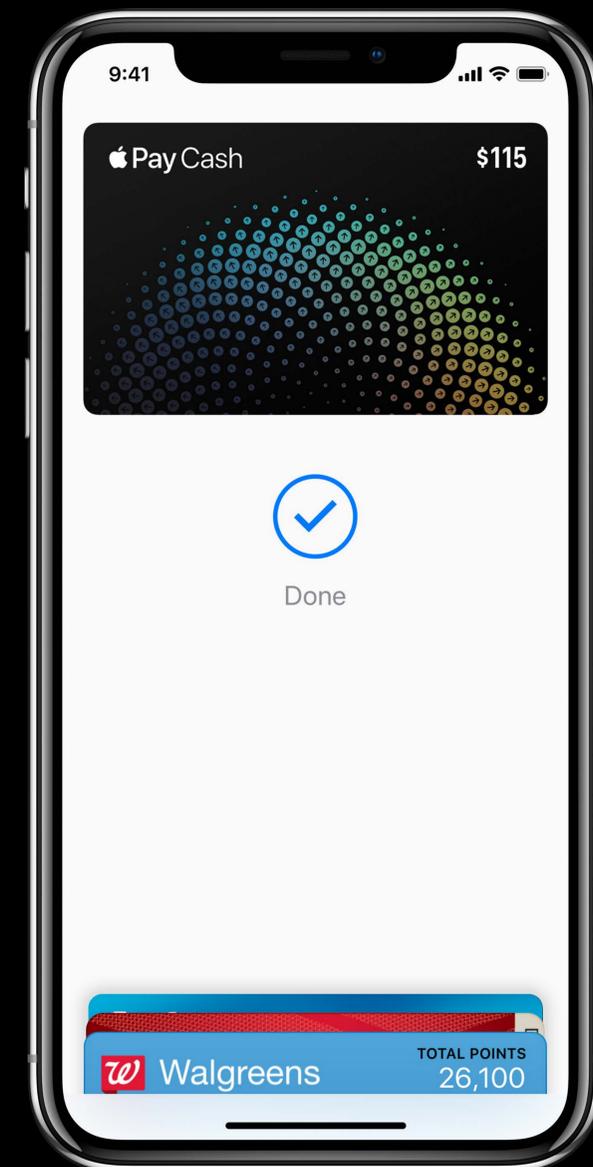
# Apple Pay Cash

Makes it easy to pay and get paid from friends and family using iMessage

Pay in-store, within apps, and on the web

Available in the U.S.

To process Apple Pay Cash payments, ensure you accept **Discover debit**

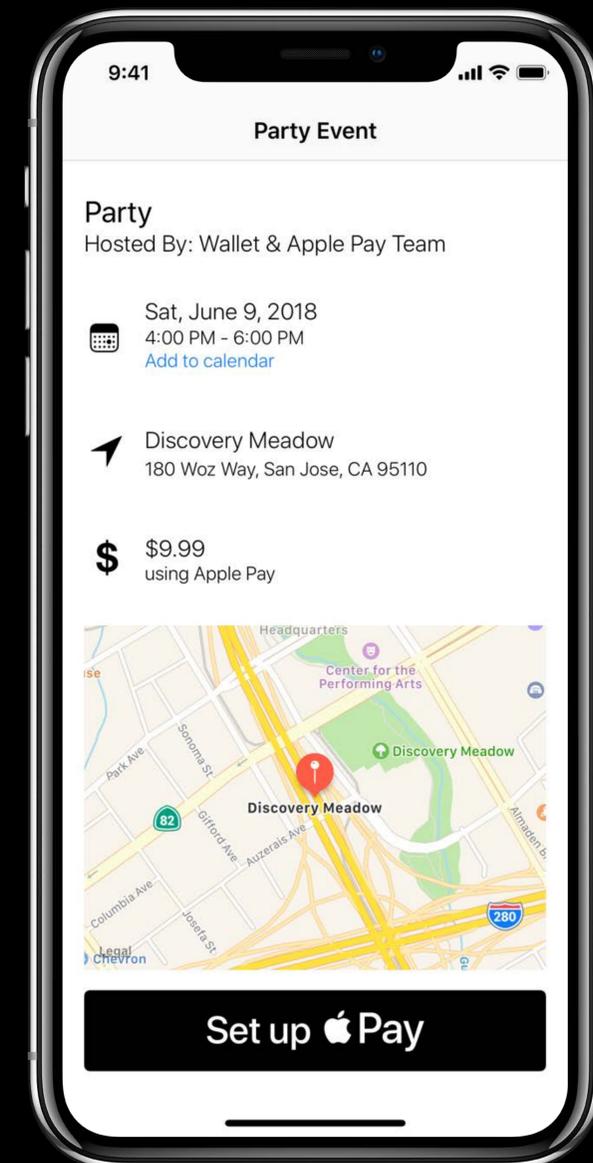


# Inline Setup

For users with no cards, set up one and return to purchase in one step

Always show the Apple Pay button if the device supports it

Use the `.setup` button type if the user has zero cards to make it even clearer

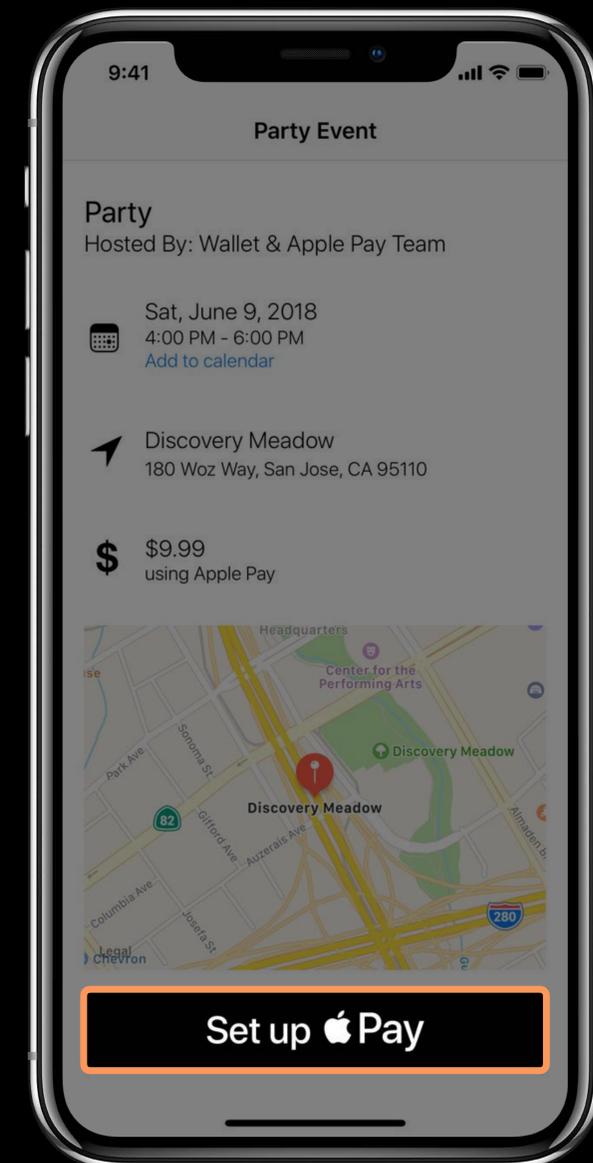


# Inline Setup

For users with no cards, set up one and return to purchase in one step

Always show the Apple Pay button if the device supports it

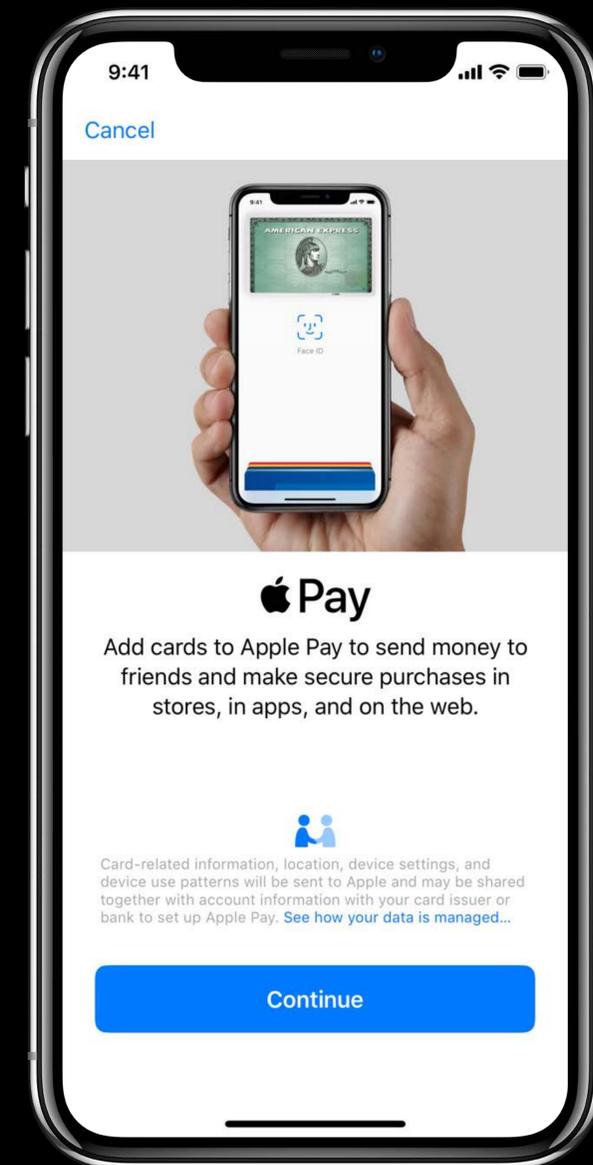
Use the `.setup` button type if the user has zero cards to make it even clearer



# Inline Setup

Apple Pay setup is presented inside your app

No additional work required

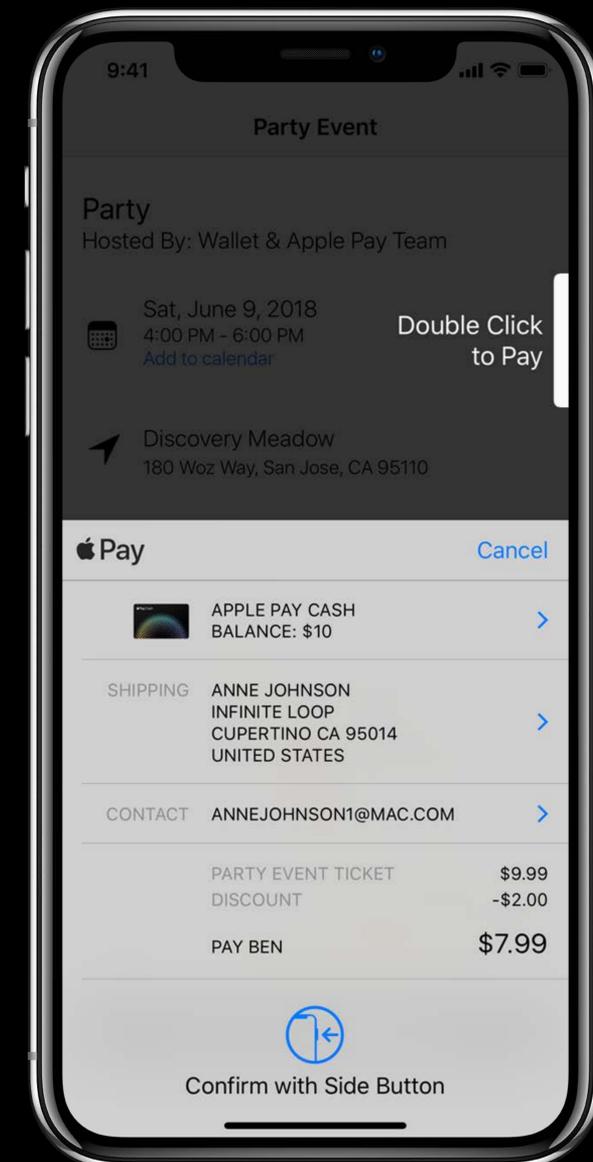


# Inline Setup

Returns the user to the purchase immediately

Automatic card activation via SMS

Faster than manual entry

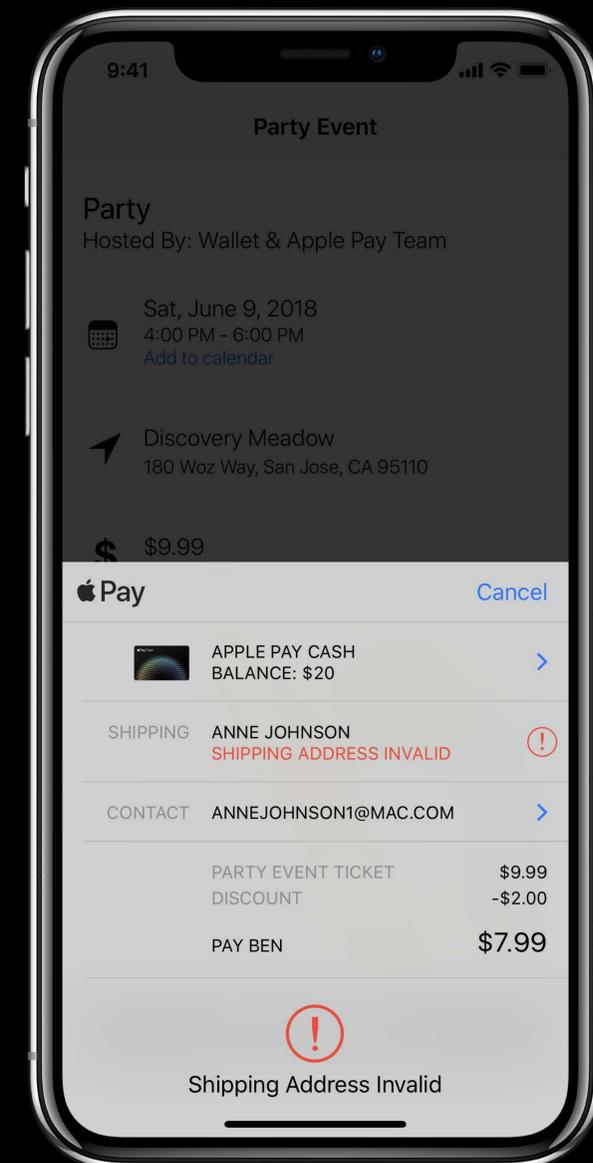


# Error Handling

Control over nonfatal errors in the  
Apple Pay Sheet

Provide your own custom error messages

Resolve issues for higher conversion



# Error Handling Tips

Important to expect fuzzy data

User has one set of data in Apple Pay shared across many apps and websites

Forcing the user to conform to your business logic actually increases friction

Instead, accept a range of inputs, i.e. Zip and Zip + 4

Infer what data you can from other fields, such as **city** and **state** from **zip**



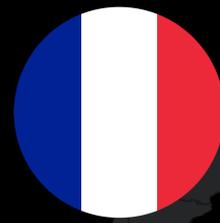
Australia



Canada



China



France



Hong Kong



Ireland



Italy



Japan



New Zealand



Russia



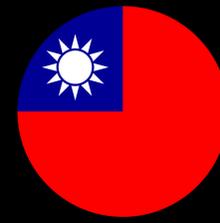
Singapore



Spain



Switzerland



Taiwan



U.K.



U.S.



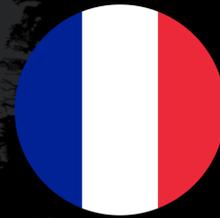
Australia



Canada



China



France



Hong Kong



Ireland



Italy



Japan



New Zealand



Russia



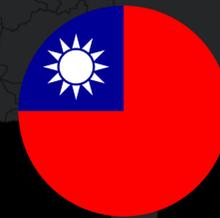
Singapore



Spain



Switzerland



Taiwan



U.K.



U.S.



Ukraine



Denmark



Finland



Sweden



Brazil



UAE



China



**Great Experiences with Apple Pay**

**Easier**

**Faster**

**Clearer**

**Easier** with Upstreaming

**Faster**

**Clearer**

**Easier** with Upstreaming

**Faster** with Defaulting

**Clearer**

**Easier** with Upstreaming

**Faster** with Defaulting

**Clearer** with Streamlining



FREE 3-DAY SHIPPING & FREE RETURNS

HOME



### MEN'S MORPH JACKET

\$249.00

★★★★★ 104 REVIEWS

SEE PRODUCT DETAILS

URBAN NAVY



SIZE



FIT FINDER FIND YOUR SIZE

FREE 3-DAY SHIPPING

ADD TO CART



GET IT IN A TNF STORE

VIPeak MEMBERS EARN 2,490 PEAKPOINTS LEARN MORE



#### LIFETIME GUARANTEE

Our products are fully warranted against defects in materials and workmanship with a lifetime guarantee. Exclusions apply\*. SEE DETAILS



#### FREE RETURNS



9:41



MY BAG (2)

EDIT



**BARNEYS NEW YORK**  
Large Weekender Bag  
Size: One Size Qty: 1  
Only 1 remaining

\$195



**BARNEYS NEW YORK**  
Flat Card Case  
Size: One Size Qty: 1

\$115

Add Gift Wrap

Have Questions?

Estimated Total

\$310.00

Pay

PROCEED TO CHECKOUT

# Apple Pay Button

Localized in all device languages

Available in a number of styles and colors

Scalable for all supported devices

You'll get our future updates without any work

Available in the SDK from iOS 8.3 and WebKit from iOS 10.1/macOS 10.12.1

# Book with Apple Pay

NEW

```
let button = PKPaymentButton.init(paymentButtonType: .book, paymentButtonStyle: .white)
```

Book with  Pay

Use when booking a hotel or with a ridesharing service

# Subscribe with Apple Pay

NEW

```
let button = PKPaymentButton.init(paymentButtonType: .subscribe, paymentButtonStyle: .white)
```

Subscribe with  Pay

Use for subscription-based purchases

# Checkout with Apple Pay

NEW

```
let button = PKPaymentButton.init(paymentButtonType: .checkout, paymentButtonStyle: .white)
```

Check out with  Pay

Use if you have multiple references to checkout on the same page

Makes it clear to the user that checkout button performs Apple Pay

# Corner Radius

NEW

```
let button = PKPaymentButton.init(paymentButtonType: .checkout, paymentButtonStyle: .white)
button.cornerRadius = 0
```



Check out with  Pay

Modify this to match the style of your existing app or website

Will fallback to use the default rounded corner on unsupported versions

# Apple Pay Button

New types and corner radius are available in iOS 12 today

Coming to WebKit in a future release

Fallback to using other types on unsupported versions

# Guest Checkout

Design for guest checkout as the first experience

Blocking first-time purchase increases friction leading to abandonment

Account creation should be desired by the user, not enforced

Create accounts post-purchase using data from Apple Pay



9:41 AM

100%

🔒 jcrew.com

SHIPPING AT

JCREW.COM

(YOUR CLOSET WILL THANK YOU TOO.)

ORDER NUMBER: 1133199050

**REGISTER NOW**

WANT TO MAKE CHECKOUT QUICKER AND EASIER NEXT TIME? JUST SIGN UP FOR A J.CREW ACCOUNT...

Password

Confirm Password

**REGISTER**



# Defaulting

Your customers will look for and expect Apple Pay

Make purchasing faster by setting Apple Pay as the default option

Apple Pay has the most up-to-date customer information

Use inline setup if the user has no cards

9:41



## Order Review

[Cancel](#)

### Pay with

 Apple Pay [>](#)

Starting Subtotal	\$214.99
<b>Extra Savings</b>	<b>\$0.00</b>
Subtotal	\$214.99
Shipping Fee ⓘ	FREE
Estimated Tax ⓘ	\$0.00
<b>Order Total</b>	<b>\$214.99</b>

Enter your code

[Apply](#)

Some rules and restrictions may apply ⓘ

### Shipping Options

The options below are tied to your delivery address.

Shipment 1 of 1

1 item fulfilled by Jet

Get it by **Tue, Jun 12**

	<b>Beats Studio3 Wireless Headphones</b> <b>\$214.99</b> Quantity: 1 Color: White
---	--

Order Total:  
**\$214.99**

**Buy with  Pay**

# Streamlining

Prefer customer information from Apple Pay

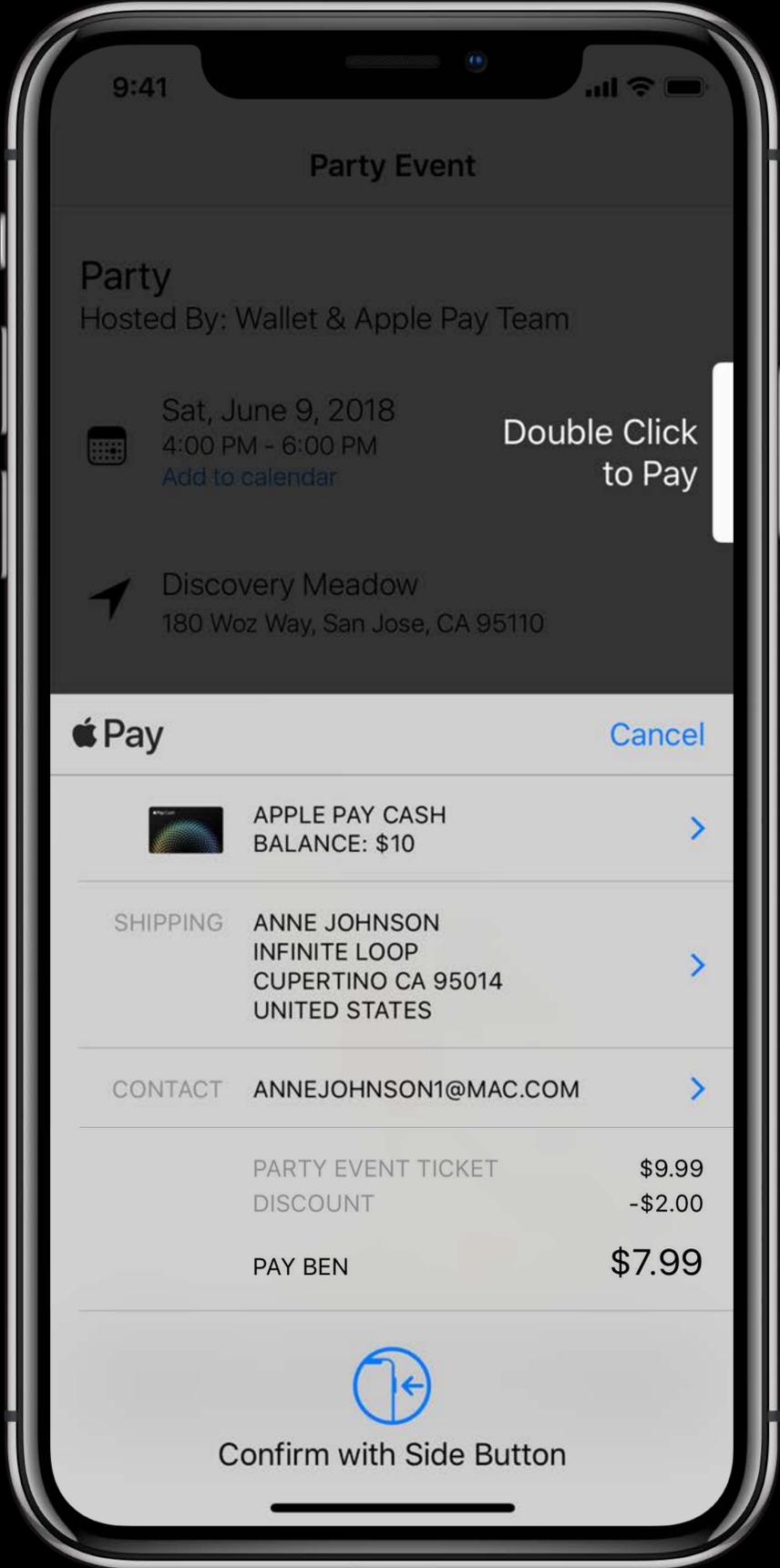
Users have contact, shipping, and billing information already set up

Request only relevant fields

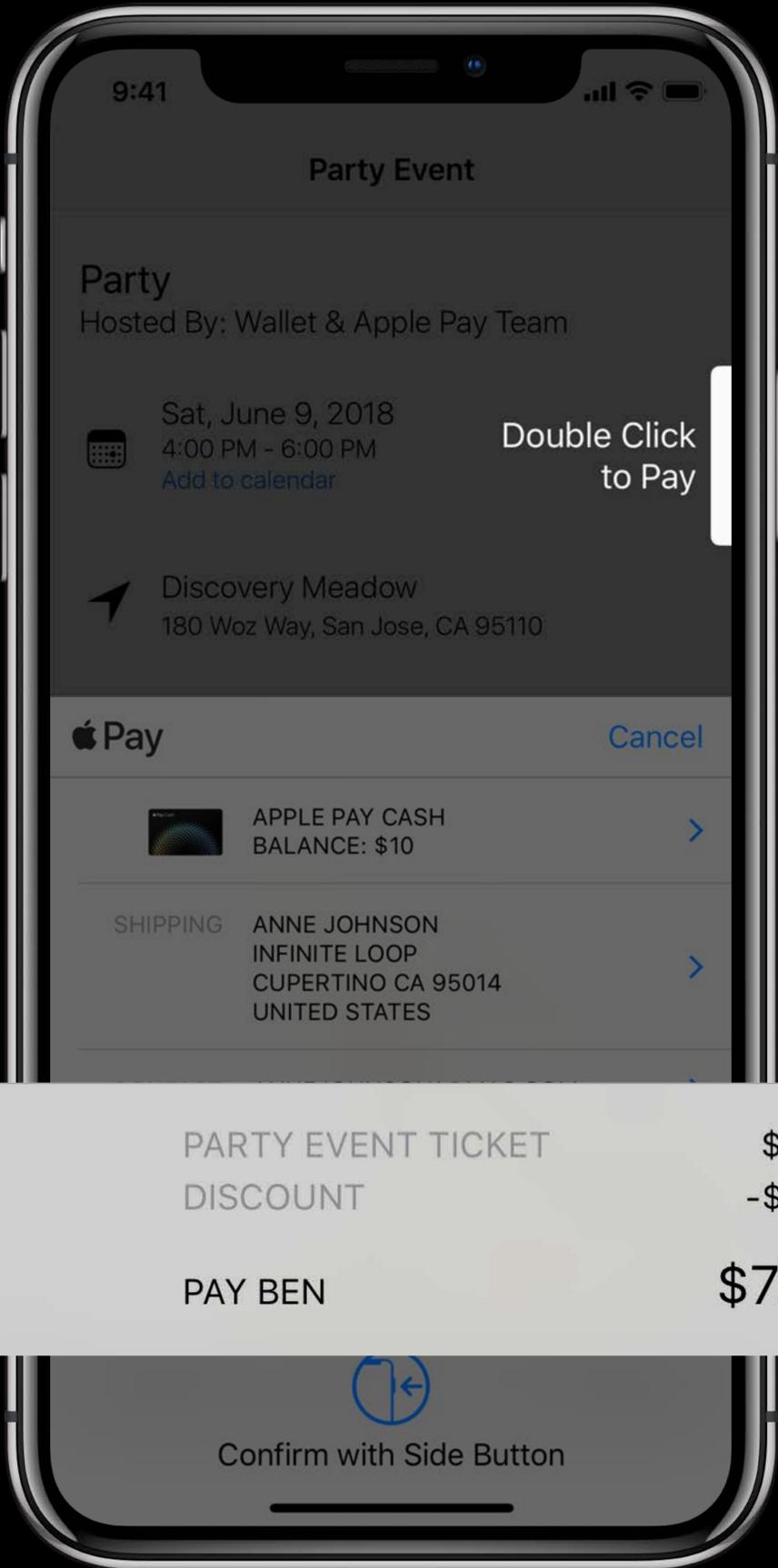
Additional fields can lead to higher abandonment

Collect other information before, checkout begins such as the number of items

NEW



NEW

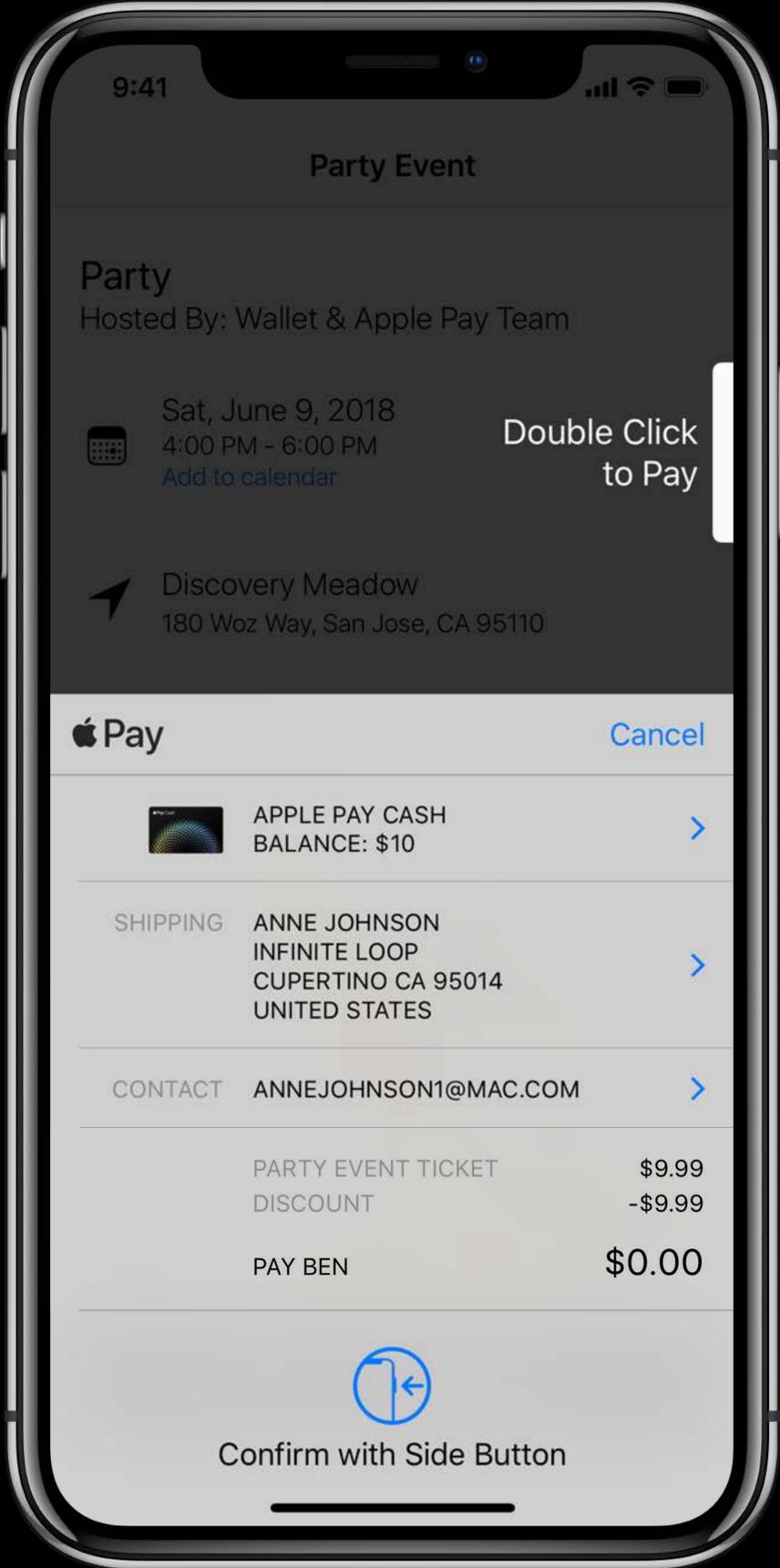


PARTY EVENT TICKET	\$9.99
DISCOUNT	-\$2.00
<b>PAY BEN</b>	<b>\$7.99</b>

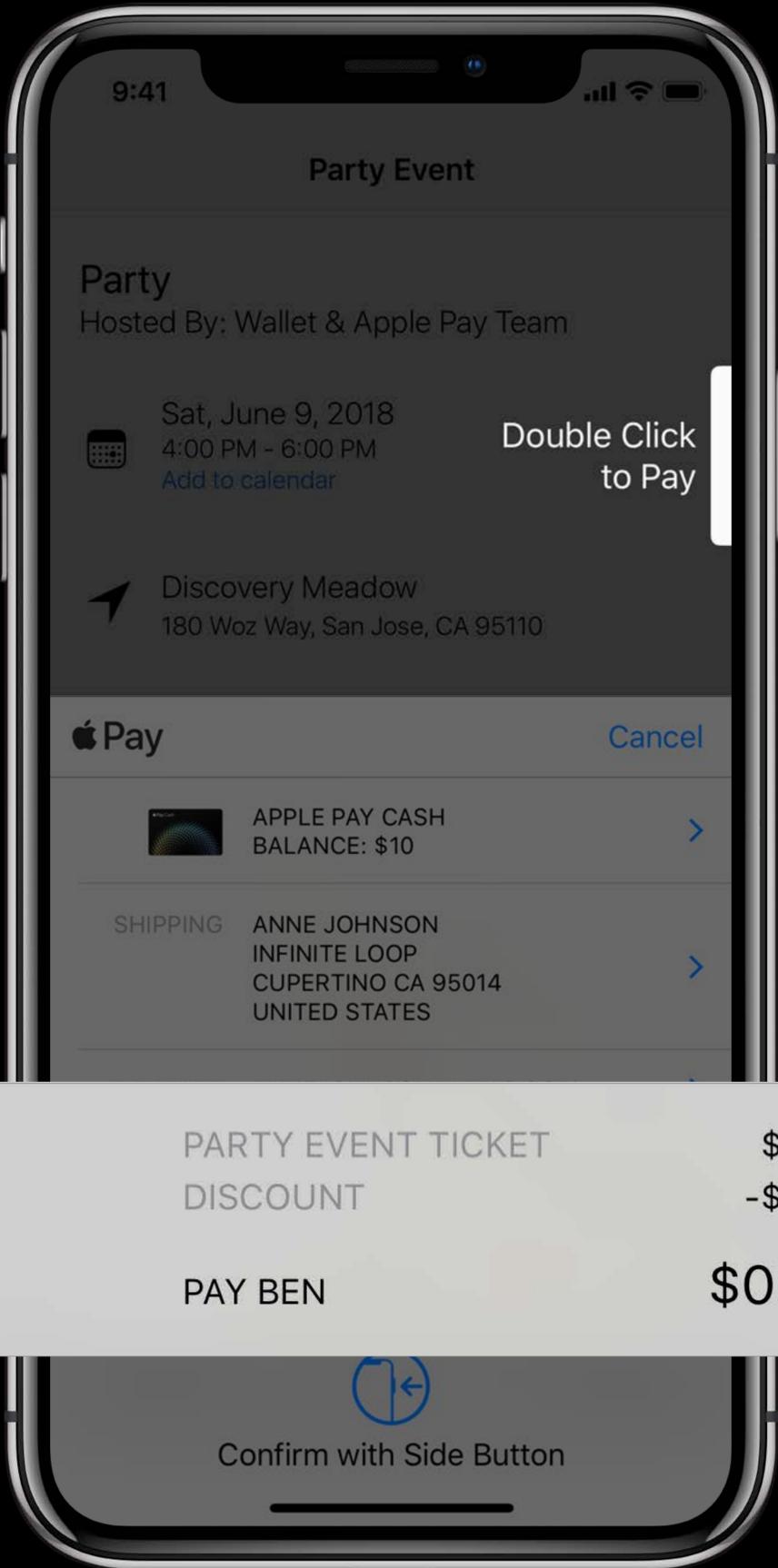


Confirm with Side Button

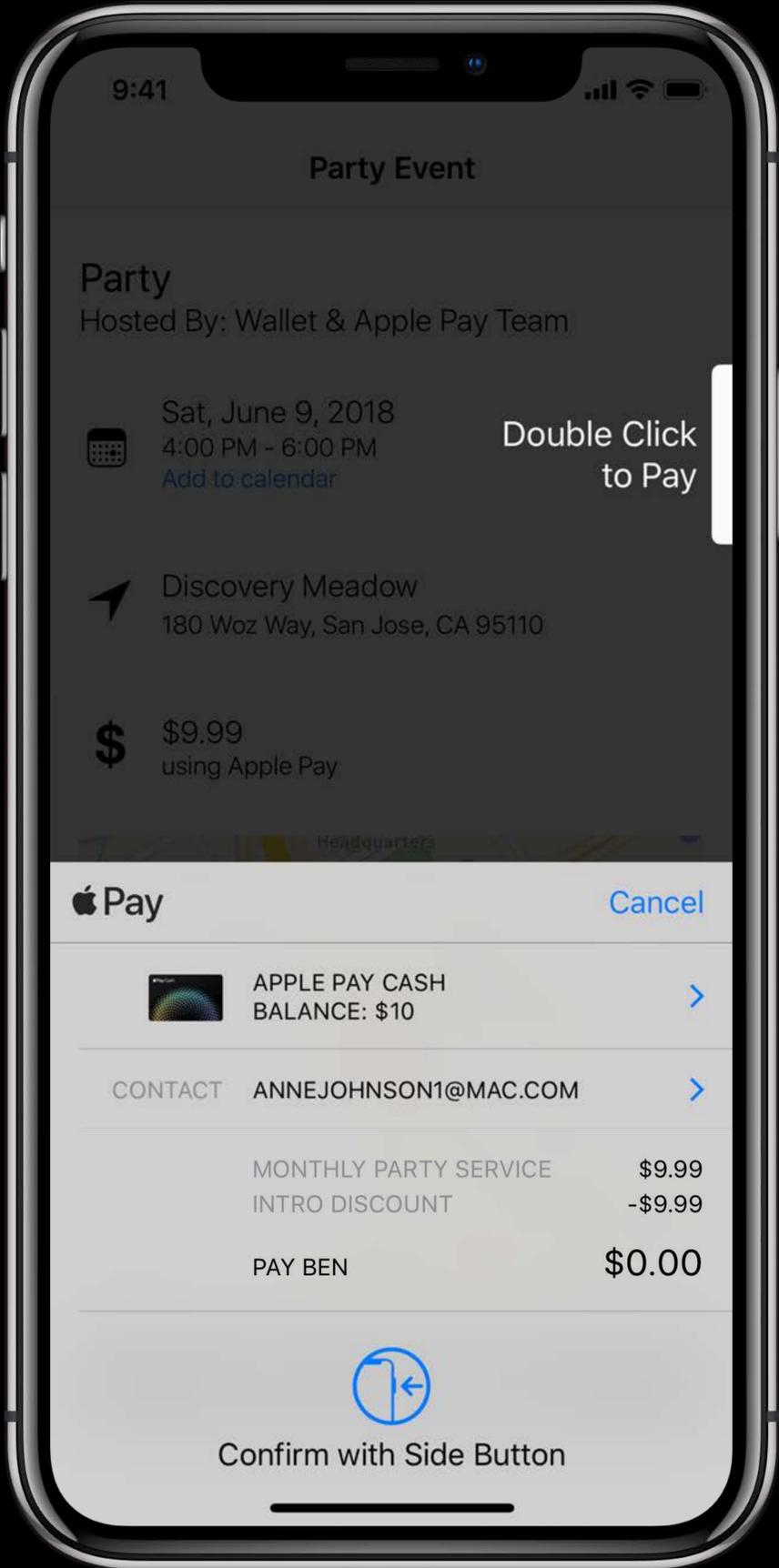
NEW



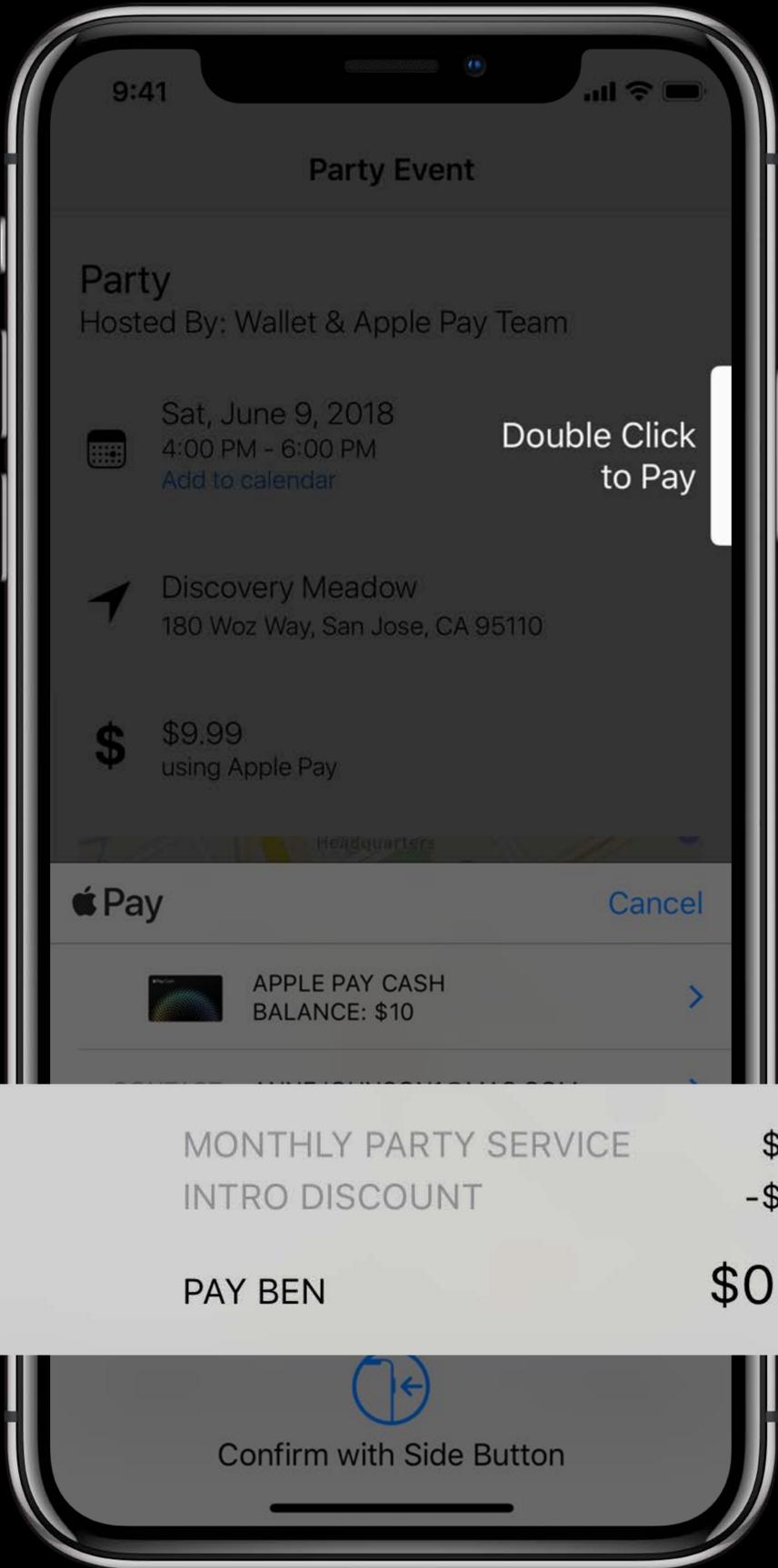
NEW



NEW



NEW



Double Click  
to Pay

Apple Pay

Cancel



APPLE PAY CASH  
BALANCE: \$10



MONTHLY PARTY SERVICE	\$9.99
INTRO DISCOUNT	-\$9.99
<b>PAY BEN</b>	<b>\$0.00</b>



Confirm with Side Button

# Zero Total Support

Available in iOS 12 today

Coming to WebKit in a future release

Make sure to fallback to existing behavior on older versions

# New Supported Networks

NEW

```
PKPaymentNetworkElectron API_AVAILABLE(ios(12.0), watchos(5.0));
```



```
PKPaymentNetworkMaestro API_AVAILABLE(ios(12.0), watchos(5.0));
```



```
PKPaymentNetworkVPay API_AVAILABLE(ios(12.0), watchos(5.0));
```



Support as many payment networks as possible for the best user experience

Available in iOS 12 today and WebKit in a future release

W3C Payment Request API  
and  
Apple Pay JS API

# Apple Pay JS API or Payment Request API

Apple Pay JS API

Payment Request API

Cross browser solution



Error handling with custom error messaging



Automatic selection for affiliated and cobranded cards



Adjust prices based on user selection of affiliated cards



Phonetic name collection



Supported from

iOS 10  
macOS 10.12

iOS 11.3  
Safari 11.1 on macOS 10.12

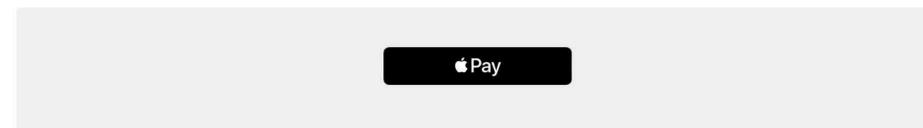
# Apple Pay on the Web Demo

On this page

- [Overview](#)
- [Requirements](#)
- [Display an Apple Pay Button](#)
- [Create a Payment Request](#)
- [Complete Merchant Validation](#)
- [Respond to Payment Sheet Interactions](#)
- [Shipping Address Selection](#)
- [Authorize the Payment](#)
- [Demo Transaction Transcript](#)
- [Additional Resources](#)
- [Questions or Feedback](#)

## Overview

Try an Apple Pay test transaction below. **Your card will not be charged.**



You can use this page to learn how to enable Apple Pay on the web using [Apple Pay JS](#). Each button is preconfigured with some default values, and if you want to explore further you can modify values in the code blocks throughout the page to customize different payment sheet experiences. A transcript of server responses is displayed at the bottom of the page after each transaction for context.

## Requirements

This demo is built using Apple Pay JS version 3, and requires the following:

- iOS devices running iOS 11 or later
- Safari 11 on Mac devices running macOS 10.13 or later

## Getting Started

This guide assumes you have already [set up your environment](#) to process Apple Pay

BUTTON STYLE

Black (Default) ▾

BUTTON TYPE

Buy ▾

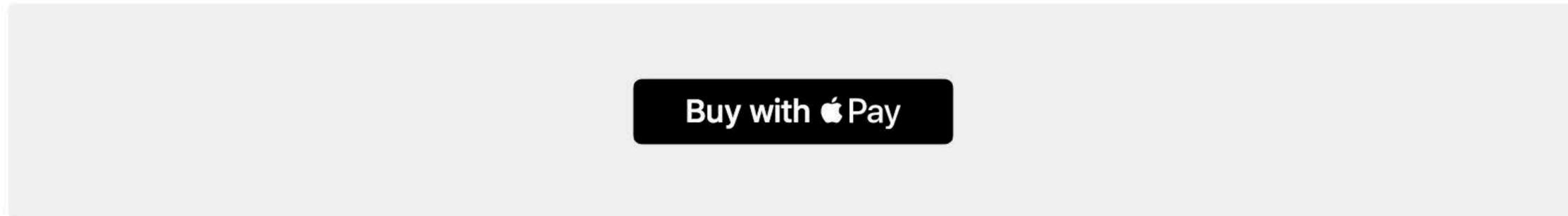
LANGUAGE

EN ▾

Width: 150px



Height: 30px



```
style="-webkit-appearance: -apple-pay-button;  
      -apple-pay-button-type: buy;  
      -apple-pay-button-style: black;  
lang=en
```

## Try it: ApplePayPaymentRequest

Configure the values in the `ApplePayPaymentRequest` structure below to see how the information is displayed on the payment sheet. Choose Basic Request to see a payment sheet with only the required fields. Choose Detailed Request to include optional fields.

- Basic Request (required fields only)
- Detailed Request

```
1 {  
2   "countryCode": "US",  
3   "currencyCode": "USD",  
4   "merchantCapabilities": [  
5     "supports3DS"  
6   ],  
7   "supportedNetworks": [  
8     "visa",  
9     "masterCard",  
10    "amex",  
11    "discover"  
12  ],  
13  "total": {  
14    "label": "My Store (Card is not charged)",  
15    "type": "final",  
16    "amount": "5.00"  
17  }  
18 }
```

Ln: 16 Col: 20



# More Information

<https://applepaydemo.apple.com>

# Getting the Most from Passes

# Why Use Passes?

Easy to use contactlessly or with a barcode

Synced across all devices and backed with iCloud

Intelligently shown on lock screen and in search for quick access

Continue the seamless Apple Pay experience into the physical world

Adding passes to Wallet

Adding passes to Wallet

Making passes look great

Adding passes to Wallet

Making passes look great

Rich pass content

Adding passes to Wallet

Making passes look great

Rich pass content

Contactless passes

# Automatically Adding Passes

```
// PKPassLibrary

func addPasses(_ passes: [PKPass], withCompletionHandler completion:
((PKPassLibraryAddPassesStatus) -> Swift.Void)? = nil)
```

Presents a simple alert to the user requesting to add or review

Less friction than presenting `PKAddPassesViewController` directly

Handle the review status in the callback

# Adding Passes Best Practices

Suggest adding passes that were created outside of your app

Add related passes to Wallet as a group

Make it easy for people to quickly add passes they do not have

Let people jump to their passes in Wallet from your app

# Designing Passes

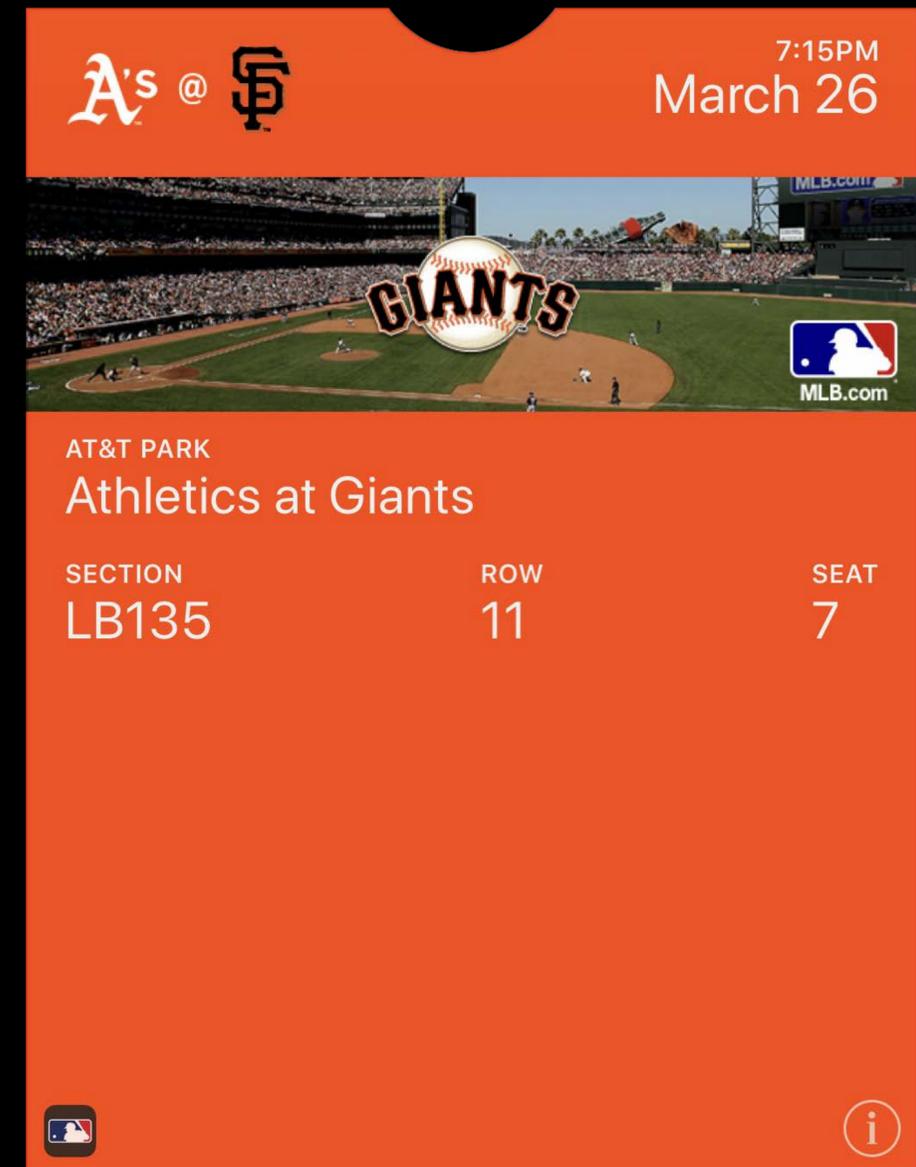
Use pass fields to display relevant text

Use vibrant colors to make your pass stand out

Design a pass that looks great on all devices

Avoid reproducing existing physical passes

Don't encode user information in the strip image

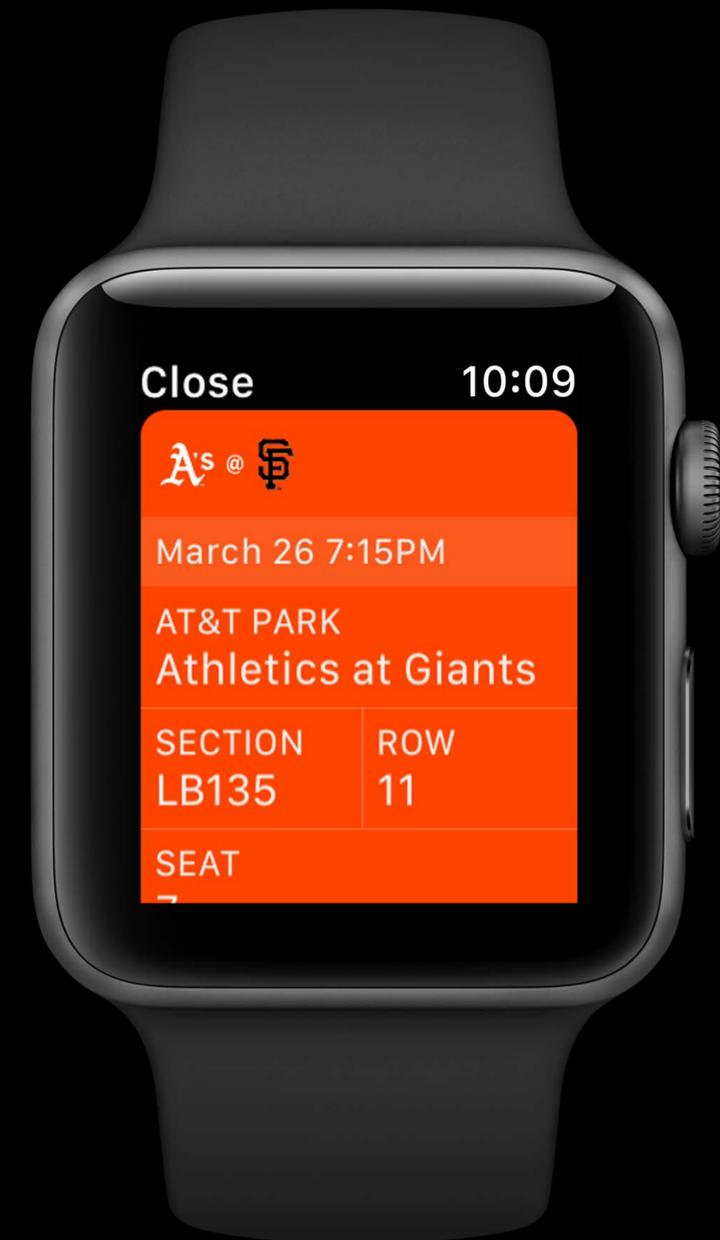


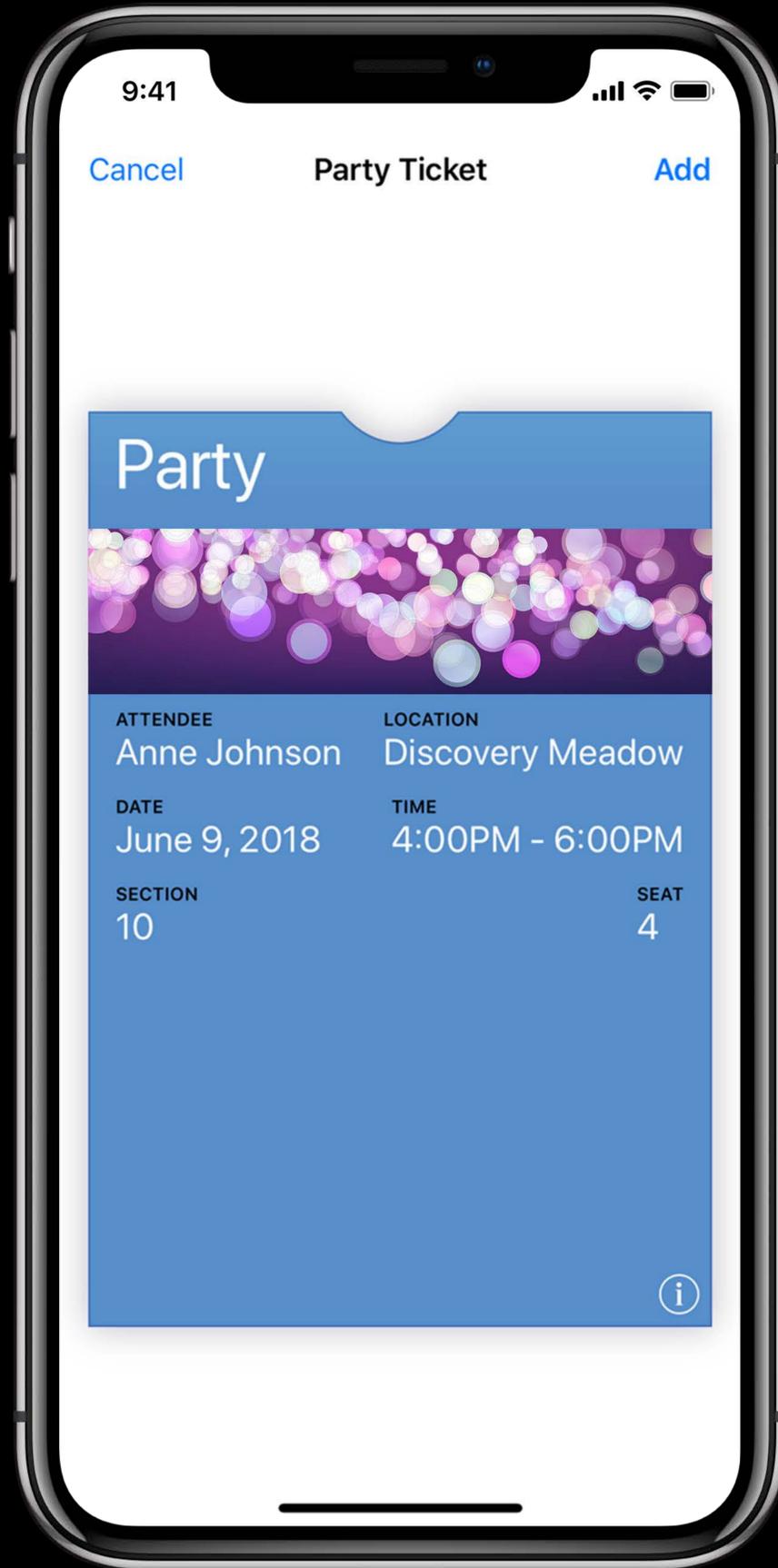
# Passes on Apple Watch

Does not support the strip image

Thumbnail image is not displayed

Users cannot access the pass details





NEW

```
"auxiliaryFields": [  
  {  
    "label": "Date",  
    "key": "Date",  
    "value": "June 9, 2018",  
    "row": 0  
  },  
  {  
    "label": "Section",  
    "key": "Section",  
    "value": "10",  
    "row": 1  
  }  
]
```

# Additional Row Support

“Row” can only be used in auxiliary fields in an event ticket type pass

Only values of 0 and 1 supported

Coming in a future seed

On older versions, “row” is ignored

Auxiliary fields are then displayed on one row up to the limit of fourth fields per row

Consider this when designing your passes

**Rich Pass Content**

# Relevancy

Add this functionality with the "locations", "relevantText", and "relevantDate" pass JSON fields

Pass appears on lock screen at the right moment

Handles multiple relevant passes

Always add relevancy information and trust the system to present as required



NEW

# Semantic Tags

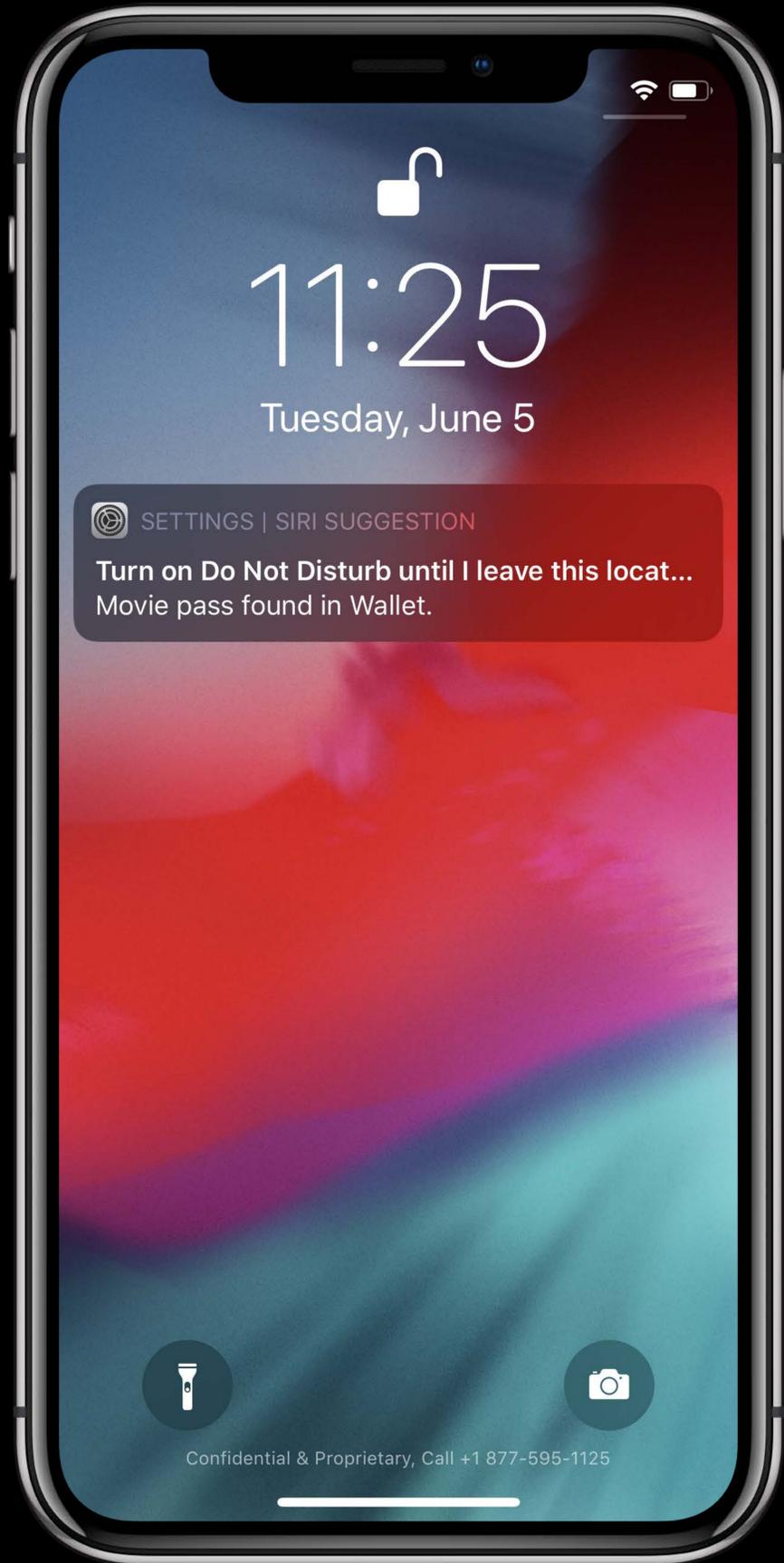
```
{
  "key": "event",
  "label": "Apple Park",
  "value": "Revenge Of The Passes",
},
{
  "key": "runtime",
  "label": "RUNTIME",
  "value": "2:45",
},
{
  "key": "theater",
  "value": "F5",
  "label": "THEATRE",
}
```

```
{  
  "key": "event",  
  "label": "Apple Park",  
  "value": "Revenge Of The Passes",  
  "semantics": {  
    "eventName": "Revenge Of The Passes: A Wallet Story",  
    "venueName": "Apple Park 1",  
    "venuePhoneNumber": "+1(408)888-8888"  
  }  
}
```

```
{
```

```
  "semantics": {  
    "eventType": "PKEventTypeMovie",  
    "silenceRequested": true,  
    "duration" : 7245  
  }
```

```
}
```



11:25

Tuesday, June 5



SETTINGS | SIRI SUGGESTION

Turn on Do Not Disturb until I leave this locat...  
Movie pass found in Wallet.



Confidential & Proprietary, Call +1 877-595-1125

# Semantic Tags

Great way to add machine readable information to passes

70+ event and transit type tags supported

Supported semantic keys will be available in online documentation soon

Relevancy information works in combination with semantic tags

# Contactless Passes



# Contactless Passes

Contactless and relevancy for lock screen access feels like magic

Requires an **NFC Certificate** to get started

# Contactless Passes

Contactless and relevancy for lock screen access feels like magic

Requires an **NFC Certificate** to get started

**Contact us** for access: <https://developer.apple.com/contact/passkit>

Your readers must support the Apple Value added Services protocol

**Verifone**



**EQUINOX**



**IDTECH**  
Value through Innovation



9:41



Duke



Molly Wiebe  
Student

FLEX BALANCE  
\$60

DINING POINTS  
\$650



Apple Pay Cash

\$124

**ASSA ABLOY**

**Blackboard®**

# *Demo*

Katie Calabro, Apple Pay and Wallet Software Engineer

# Digital to Physical Commerce

9:41



Browse



# Apple Park

[More Info](#)

[Other Locations](#)

Today's Specials  
Pizza



Prosciutto con vegetali tarufati  
\$9.00

♥ My Favorites >

🍴 Main Stations >



Browse



Pay



My Info

9:41



Browse



# Apple Park

[More Info](#)

[Other Locations](#)

Today's Specials  
Pizza



Prosciutto con vegetali tarufati  
\$9.00

Buy with Pay

♥ My Favorites >



Browse



Pay



My Info

9:41



[← Back](#)

Pizza



**SPECIAL**

**Prosciutto con vegetali tarufati**

**\$9.00**

Fresh mozzarella and fontina cheese, prosciutto, roma tomatoes, basil, and extra virgin-olive oil.

Add to Order

Buy with Pay



Browse



Pay



My Info

9:41



Browse



# Apple Park

[More Info](#)

[Other Locations](#)

Today's Specials  
Pizza

Double Click  
to Pay



Apple Pay

[Cancel](#)



GREEN CARD  
(•••• 1234)

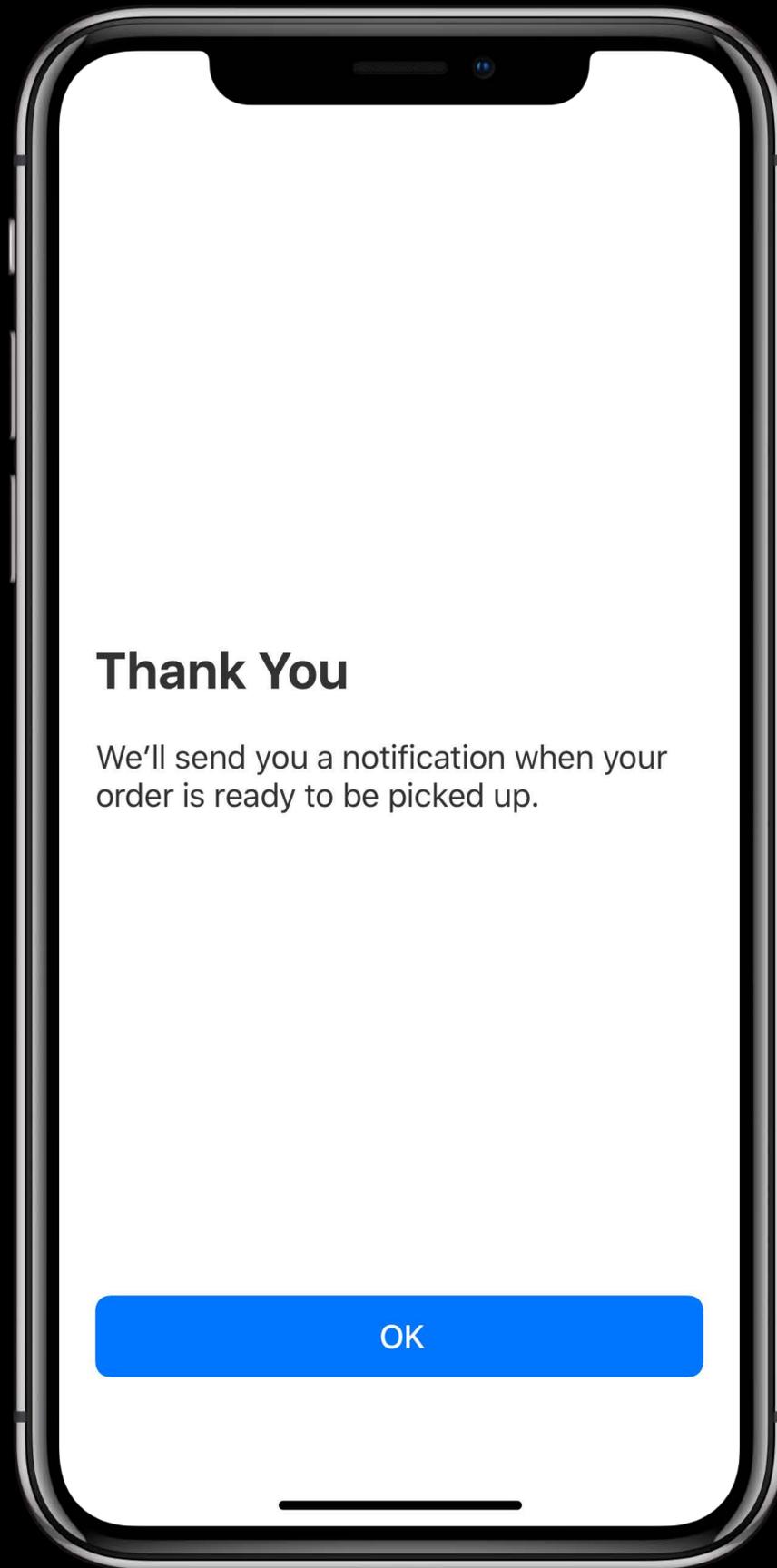


PAY CAFFÈ MACS

\$9.00



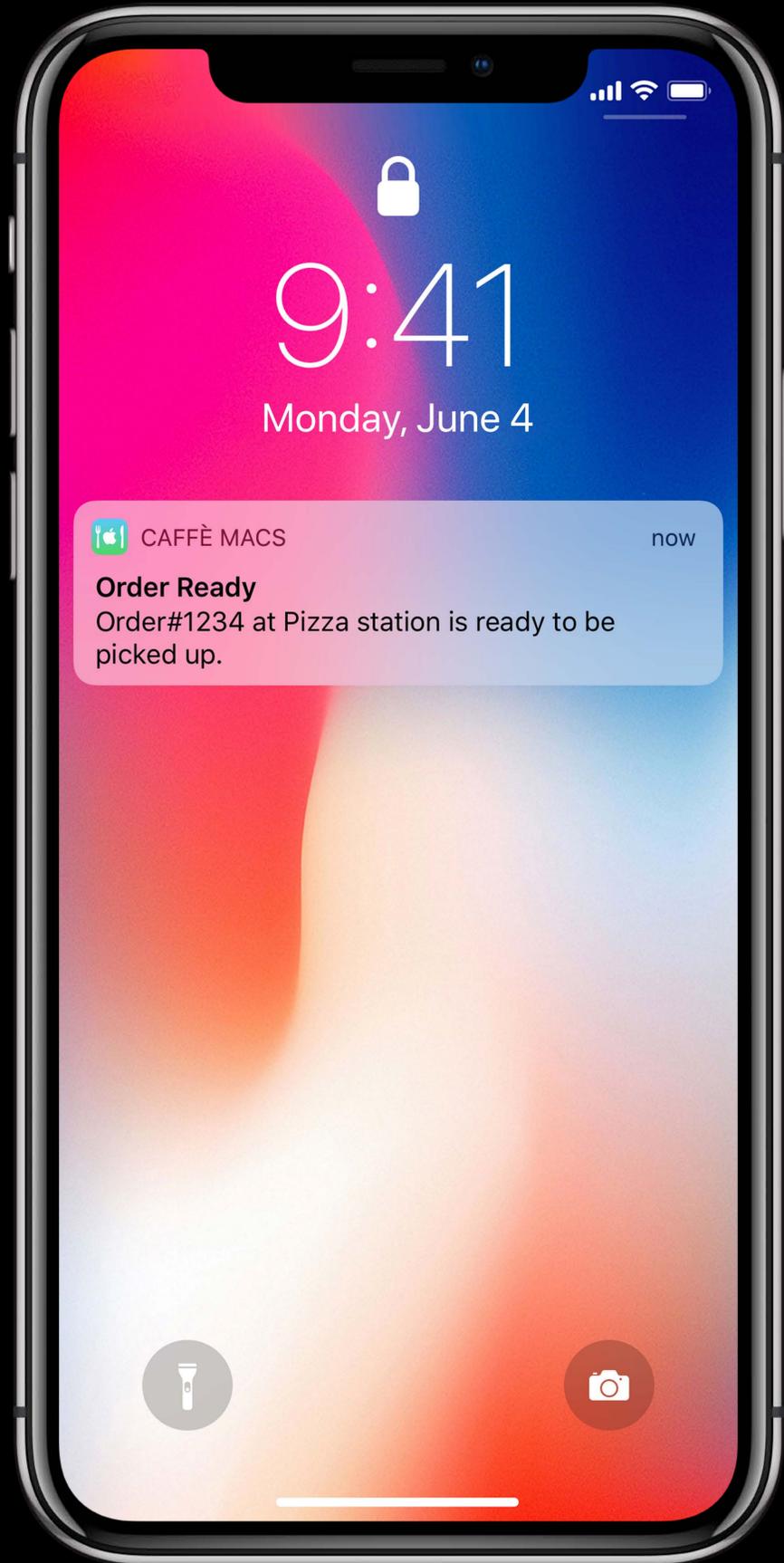
Confirm with Side Button



## Thank You

We'll send you a notification when your order is ready to be picked up.

OK



9:41

Monday, June 4



CAFFÈ MACS

now

**Order Ready**

Order#1234 at Pizza station is ready to be picked up.



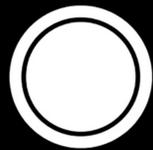
大众点评二维码

在“大众点评”中打开



Apple Pay 选美团闪付更优惠

摄影 慢动作 视频 照片 人像 正方形 全景



9:41



大壶春(旗舰店)



大壶春(旗舰店)

欢迎光临本店, 祝您用餐愉快!

生煎类

生煎类

汤类

点心类

菜饭

豆浆饮料



网友推荐 鲜肉生煎  
👍 999+ 销量 6065  
¥8



鲜肉生煎半两  
👍 1 销量 797  
¥4



网友推荐 蛤蜊鲜肉生煎  
👍 999+ 销量 673  
¥12



网友推荐 大虾鲜肉生煎  
👍 999+ 销量 1472  
¥12



网友推荐 秘制小锅生煎  
👍 461 销量 1335  
¥31



网友推荐 小虾鲜肉生煎  
👍 999+ 销量 1472  
¥12



9:41



购物车(3)

就餐人数

1人用餐 >

共3份菜品

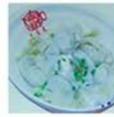
🗑️ 清空



鲜肉生煎半两

¥4

⊖ 1 ⊕



菜肉大馄饨

¥13

⊖ 1 ⊕



原味豆浆

¥5

⊖ 1 ⊕

总计: ¥22

确认下单

9:41

← 相机



支付订单

支付剩余时间 29:55

¥ 22.00

美团点评点餐订单-060632622781

Apple Pay

9:41



支付订单

支付剩余时间 25:18

¥ 22.00

连接侧边按钮  
以付款

美团点评点餐订单-060632622781

Apple Pay

取消



美团闪付  
(•••• 8447)



付给：大众点评网

¥22.00



通过侧边按钮确认

# Certificates Expiry

If you're a **payment processor** or **merchant**

Make sure to check your server-side certificate expiry dates in the developer portal

They are easy to renew

Renew before expiry to avoid disruption taking Apple Pay

# More Information

<https://developer.apple.com/wallet/>

<https://developer.apple.com/apple-pay/>

 **WWDC18**