What’s New in Apple Pay & Wallet

Session 714

Nick Shearer, Software Magician
Jon Nguy, Software Engineer
The Apple Pay experience

What’s new with Apple Pay

What’s new with Wallet
The Apple Pay experience
What’s new with Apple Pay
What’s new with Wallet
More Information

Getting started with the Apple Pay API

Integrating Apple Pay with your payment processor
The Apple Pay Experience
Apple Pay
<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. MARTEN Niel Style Men's Boot Size: 9</td>
<td>$100.00</td>
</tr>
<tr>
<td>Shipping</td>
<td>$0.99</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$100.99</strong></td>
</tr>
<tr>
<td><strong>Net Charged</strong></td>
<td><strong>$100.99</strong></td>
</tr>
</tbody>
</table>

**PAYMENT**

- **Apple Pay**
- CHASE SAPPHIR...ERVE (**** 8144) 165 DAVIS STREET HILLSBOROUGH...
- **Shipping** KATE BELL 165 DAVIS STREET HILLSBOROUGH CA 94010

- **Sub Total** $100.00
- **Shipping** $0.99
- **Pay Poshmark** $100.99

Pay with Touch ID
Buy with Apple Pay
Invest in a world where everyone has clean water

100% brings clean water to people in need.

$60 USD/MONTH

Donate with Apple Pay

Give by credit card

Link your bank
“In the last year we’ve been able to reduce the time it takes to donate from around two minutes to ten seconds. Apple Pay has helped us improve the user experience, further reducing any areas of friction and streamlining the giving journey.”

Matt Jerwood, Head of Digital Fundraising
Apple Pay for Donations
Apple Pay for Donations

Accept donations for your nonprofit simply and securely
Apple Pay for Donations

Accept donations for your nonprofit simply and securely

Available within apps and on the web
Apple Pay for Donations

Accept donations for your nonprofit simply and securely

Available within apps and on the web

New donation button style
Apple Pay for Donations

Accept donations for your nonprofit simply and securely

Available within apps and on the web

New donation button style

Why Apple Pay?
Best way for users to pay
Best way for merchants to be paid
The Checkout Process
The Checkout Process

Product Page
The Checkout Process

Create Account/Sign In → Product Page
The Checkout Process

Create Account/Sign In → Product Page → Shipping Address
The Checkout Process

1. Create Account/Sign In
2. Product Page
3. Shipping Address
4. Billing Address
The Checkout Process

Create Account/Sign In → Product Page

→ Shipping Address

→ Billing Address

→ Card Details
The Checkout Process

1. Product Page
2. Create Account/Sign In
3. Shipping Address
4. Billing Address
5. Card Details
6. Review Transaction
The Checkout Process

1. Create Account/Sign In
2. Product Page
3. Shipping Address
4. Billing Address
5. Card Details
6. Review Transaction
7. Confirmation
The Checkout Process

Create Account/Sign In ➔ Product Page ➔ Buy with Apple Pay

Shipment Address ➔ Billing Address ➔ Card Details ➔ Review Transaction ➔ Confirmation
Apple Pay Performance
Apple Pay Performance

2x to 5x increased conversions with Apple Pay
Apple Pay Performance

2x to 5x increased conversions with Apple Pay

Higher purchase frequency
Apple Pay Performance

2x to 5x increased conversions with Apple Pay

Higher purchase frequency

Faster checkouts
Apple Pay Performance

2x to 5x increased conversions with Apple Pay

Higher purchase frequency

Faster checkouts

Improved customer acquisition and satisfaction
Invoices with Apple Pay paid 30% faster
50% of invoices paid immediately
“We’ve seen a 250% increase in our conversion rate with Apple Pay”

Bill Devine, Director of Product at Indiegogo
Other Benefits
Other Benefits

Reduction in chargebacks
Other Benefits

Reduction in chargebacks

No need to handle or store credit card numbers
Other Benefits

- Reduction in chargebacks
- No need to handle or store credit card numbers
- Trusted user experience
Other Benefits

Reduction in chargebacks

No need to handle or store credit card numbers

Trusted user experience
Perfecting the Apple Pay Experience
Perfecting the Apple Pay Experience

Make purchasing easier
Perfecting the Apple Pay Experience

Make purchasing easier

Make purchasing faster
Perfecting the Apple Pay Experience

Make purchasing easier
Make purchasing faster
Make purchasing clearer
Perfecting the Apple Pay Experience

Make purchasing easier with upstreaming

Make purchasing faster

Make purchasing clearer
Perfecting the Apple Pay Experience

Make purchasing easier with upstreaming
Make purchasing faster with defaulting
Make purchasing clearer
Perfecting the Apple Pay Experience

Make purchasing easier with upstreaming
Make purchasing faster with defaulting
Make purchasing clearer with streamlining
Upstreaming

Accelerate single-item purchases with Apple Pay on product pages

Accelerate multi-item purchases with express checkout
Upstreaming
Upstreaming

Placing the Apple Pay button on product pages increases conversions.

Allow purchasing without an account for best performance.
Upstreaming and Accounts
Upstreaming and Accounts

Sometimes guest checkout isn’t feasible
Upstreaming and Accounts

Sometimes guest checkout isn’t feasible

Make account creation seamless with iOS and macOS
Upstreaming and Accounts

Sometimes guest checkout isn’t feasible

Make account creation seamless with iOS and macOS

Shared web credentials make app account setup faster
Upstreaming and Accounts

Sometimes guest checkout isn’t feasible

Make account creation seamless with iOS and macOS

Shared web credentials make app account setup faster

UITextContentType helps iOS autofill your input fields
Upstreaming and Accounts

Sometimes guest checkout isn’t feasible

Make account creation seamless with iOS and macOS

Shared web credentials make app account setup faster

`UITextContentType` helps iOS autofill your input fields

Create account post-purchase using Apple Pay data
Defaulting

Make Apple Pay the default payment option where possible

Take advantage of the new in-line setup in iOS 11
Streamlining
Streamlining

Make purchasing with Apple Pay quick and simple
Streamlining

Make purchasing with Apple Pay quick and simple

Collect necessary information before checkout begins
Streamlining

Make purchasing with Apple Pay quick and simple
Collect necessary information before checkout begins
Prefer customer information from Apple Pay
Delivery & Pickup

iPad Wi-Fi 32GB - Gold $329.00
Recycle fee $5.00

Apple Pay

CHASE SAPPHIRE...ERVE (**** 8144)
165 DAVIS STREET HILLSBOROUGH...

SHIPPING
KATE BELL
165 DAVIS STREET
HILLSBOROUGH CA 94010

METHOD
FREE 2 BUSINESS DAY SHIPPING
DELIVERS: SAT JUN 3

CONTACT
KATE-BELL@MAC.COM
(555) 564-8583

BAG SUBTOTAL $334.00
FREE SHIPPING $0.00
ESTIMATED TAX $29.61
PAY APPLE $363.61

Pay with Touch ID
Payment Sheet Guidelines

General advice
Payment Sheet Guidelines

General advice

Only present and request essential information
Payment Sheet Guidelines

General advice

Only present and request essential information

Keep line items short
Payment Sheet Guidelines

General advice

Only present and request essential information

Keep line items short

Use line items to explain additional charges or show discounts
Delivery & Pickup

iPad Wi-Fi 32GB - Gold $329.00
Recycle fee $5.00

Apple Pay

CHASE SAPPHIREERVE (**** 8144)
165 DAVIS STREET HILLSBOROUGH...

SHIPPING
KATE BELL
165 DAVIS STREET
HILLSBOROUGH CA 94010

METHOD
FREE 2 BUSINESS DAY SHIPPING
DELIVERS: SAT JUN 3

CONTACT
KATE-BELL@MAC.COM
(555) 564-8583

BAG SUBTOTAL $334.00
FREE SHIPPING $0.00
ESTIMATED TAX $29.61
PAY APPLE $363.61
Payment Sheet Guidelines
Pending items

Clearly disclose when additional costs may be incurred

For example, if a gratuity can be added post-purchase

```swift
let gratuity = PKPaymentSummaryItem(label: "Optional Gratuity",
                                      amount: NSDecimalNumber(string: "5.50"),
                                      type: .pending)
```
Payment Sheet Guidelines

Total amount

Total amount must be non-negative (but may be pending)

Use the business name that will appear on the card statement
Clarify subscription details prior to payment
Include line items that reiterate billing frequency
For variable subscriptions, use pending items
WWDC Toast
Avocado Toast Subscription
$29.99/week

Enjoy bespoke, artisanal, and delicious avocado toast every day of the week, with our weekly subscription package.

Buy with Apple Pay

Apple Pay
CHASE SAPPHIRE
(•••• 5555)

DELIVERY
NICK SHEARER
2 INFINITE LOOP
CUPERTINO CA 95104
UNITED STATES

WEEKLY SERVICE $29.99
INTRO DISCOUNT -$10.00
PAY WWDC TOAST INC $19.99

Pay with Touch ID
WWDCToast
Avocado Toast Subscription
$29.99/week

Enjoy bespoke, artisanal, and delicious avocado toast every day of the week, with our weekly subscription package.

Buy with Apple Pay

<table>
<thead>
<tr>
<th>Apple Pay</th>
<th>Cancel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHASE SAPPHIRE (**** 5555)</td>
<td>&gt;</td>
</tr>
<tr>
<td>DELIVERY NICK SHEARER 2 INFINITE LOOP CUPERTINO CA 95104 UNITED STATES</td>
<td>&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEEKLY SERVICE</td>
<td>$29.99</td>
</tr>
<tr>
<td>INTRO DISCOUNT</td>
<td>-$10.00</td>
</tr>
<tr>
<td>PAY WWDC TOAST INC</td>
<td>$19.99</td>
</tr>
</tbody>
</table>

Pay with Touch ID
Payment Sheet Guidelines
Variable subscriptions

Use pending amounts for variable subscriptions

Only show a payment sheet where a change results in additional fees

When the user decreases the cost, authorization is not necessary
Apple Pay Buttons
Apple Pay Buttons

Use built-in buttons—don’t create your own
Apple Pay Buttons

Use built-in buttons—don’t create your own

Localized and scalable in a variety of styles
Apple Pay Buttons

Use built-in buttons—don’t create your own
Localized and scalable in a variety of styles
Available in the SDK from iOS 8.3
Apple Pay Buttons

Use built-in buttons—don’t create your own

Localized and scalable in a variety of styles

Available in the SDK from iOS 8.3

Simplified CSS styles for WebKit from iOS 10.1/macOS 10.12.1
// Swift
let button = PKPaymentButton(type: .buy, style: .black)
// Swift
let button = PKPaymentButton(type: .buy, style: .black)

// HTML
<button lang="en"
    style="-webkit-appearance: -apple-pay-button;
            -apple-pay-button-type: buy;"></button>
// Swift
let button = PKPaymentButton(type: .buy, style: .black)

// HTML
<button lang="en"
    style="-webkit-appearance: -apple-pay-button;
            -apple-pay-button-type: buy;">"
// Swift
let button = PKPaymentButton(type: .buy, style: .black)

// HTML
<button lang="en"
    style="-webkit-appearance: -apple-pay-button;
           -apple-pay-button-type: buy;">" />
</button>
// Swift
let button = PKPaymentButton(type: .buy, style: .black)

// HTML
<button lang="en"
    style="-webkit-appearance: -apple-pay-button;
    -apple-pay-button-type: buy;"></button>
```swift
let button = PKPaymentButton(type: .buy, style: .black)
```

```html
<button lang="en" style="-webkit-appearance: -apple-pay-button; -apple-pay-button-type: buy;">
</button>
```

<table>
<thead>
<tr>
<th></th>
<th>Swift</th>
<th>CSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple Pay</td>
<td>.plain</td>
<td>plain</td>
</tr>
<tr>
<td>Buy with Apple Pay</td>
<td>.buy</td>
<td>buy</td>
</tr>
<tr>
<td>Set up Apple Pay</td>
<td>.setUp</td>
<td>set-up</td>
</tr>
<tr>
<td>Donate with Apple Pay</td>
<td>.donate</td>
<td>donate</td>
</tr>
</tbody>
</table>

// New iOS 10.2 / macOS 10.12.3
What’s New with Apple Pay
What’s New with Apple Pay

Make best-in-class conversion even better

Help users resolve common problems

Make getting started with Apple Pay fast and easy
Setup Apple Pay

Recommend showing the Apple Pay button for all Apple Pay capable devices

Choose the Setup or Buy button style that suits your needs
Inline Setup

Apple Pay setup is now offered automatically

Simply present an Apple Pay sheet to a user without cards

Users are returned to your Apple Pay purchase immediately after setup

Still faster than a typical manual checkout
Delivery Options

Apple Watch Series 2, 38mm Stainless Steel Case with White Sport Band

Ship:
- Delivers Thu Jun 1 - Free
- Delivers Today within 2 hours - $19.00

Get it Wed May 31 when you select Next Day Shipping

More options for 94010

Pay with Other Payment Options

Chat Now
By clicking the 'Apple Pay' button, you are certifying acceptance of the terms and conditions of this sale.
Delivery Options

Apple Watch Series 2, 38mm
Stainless Steel Case with White Sport Band

Ship:
- Delivers Thu Jun 1 - Free
- Delivers Today within 2 hours - $19.00
  Get it Wed May 31 when you select Next Day Shipping

More options for 94010

Apple Pay

Pay with Other Payment Options

Chat Now

By clicking the ‘Apple Pay’ button, you are certifying acceptance of the terms and conditions of this sale.
Apple Pay Cash
Apple Pay Cash

Money received in iMessages is kept in Apple Pay Cash

Can be used in-store, within apps, and on the web

If you can process credit cards, you can process Apple Pay Cash

Further information coming soon
WWDC Toast

$5.49
Delicious avocado toast delivered straight to your conference seat.

Buy with Apple Pay

Insufficient Balance
$5.49
Delicious avocado toast delivered straight to your conference seat.

Buy with Apple Pay

Apple Pay
APPLE PAY CASH - $0.00
INSUFFICIENT BALANCE

DELIVERY
NICK SHEARER
2 INFINITE LOOP
CUPERTINO CA 95104
UNITED STATES

SUBTOTAL $5.49
DELIVERY $2.99

PAY WWDC TOAST INC $8.48

Insufficient Balance
Apple Pay Cash
Sheet considerations
Apple Pay Cash
Sheet considerations

Apple Pay Cash is only available when your balance exceeds the total to pay
Apple Pay Cash
Sheet considerations

Apple Pay Cash is only available when your balance exceeds the total to pay

Make sure your total amount reflects what you intend to charge
Apple Pay Cash
Sheet considerations

Apple Pay Cash is only available when your balance exceeds the total to pay

Make sure your total amount reflects what you intend to charge

Use a pending amount instead if you aren’t sure of the total
What’s New with Apple Pay

Supported countries

Limit payment to cards from specific countries if necessary

Useful for regulatory or other compliance needs

```javascript
paymentRequest.supportedCountries = ["US", "CA"]
```
Error Handling
Accidents Will Happen

Sometimes things go wrong

Problematic to communicate this to users
Accidents Will Happen
Accidents Will Happen

Payment instrument failed to process
Accidents Will Happen

Payment instrument failed to process

Billing address didn’t match
Accidents Will Happen

Payment instrument failed to process
Billing address didn’t match
Email address was invalid
Accidents Will Happen

Payment instrument failed to process
Billing address didn’t match
Email address was invalid
Postal address had an incorrect ZIP
Accidents Will Happen

Payment instrument failed to process
Billing address didn’t match
Email address was invalid
Postal address had an incorrect ZIP
Telephone was missing an area code
<table>
<thead>
<tr>
<th>Fatal Errors</th>
<th>Non-fatal Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can’t be resolved by the user</td>
<td>Resolvable by the user</td>
</tr>
<tr>
<td>Handled outside of Apple Pay</td>
<td>Handled within Apple Pay</td>
</tr>
<tr>
<td>Non-preventable</td>
<td>Sometimes foreseeable and preventable</td>
</tr>
</tbody>
</table>
Non-Fatal Errors

Available `PKPaymentAuthorizationStatus` values are generic

No specificity

Unable to edit existing values

Not always clear what the problem is
Improved Error Handling

New guidelines to better handle user data

New UI for users to input and edit data

New API for merchants to communicate precise errors
Handling User Data

Contact information on iOS and macOS comes from many sources

Don’t expect this information to conform to your business logic
<table>
<thead>
<tr>
<th><strong>Postal Addresses</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Street</strong></td>
</tr>
<tr>
<td><strong>City</strong></td>
</tr>
<tr>
<td><strong>State / Province</strong></td>
</tr>
<tr>
<td><strong>Postal Code / ZIP</strong></td>
</tr>
<tr>
<td><strong>Country</strong></td>
</tr>
<tr>
<td><strong>Phone</strong></td>
</tr>
<tr>
<td>Street</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>City</td>
</tr>
<tr>
<td>State / Province</td>
</tr>
<tr>
<td>Postal Code / ZIP</td>
</tr>
<tr>
<td>Country</td>
</tr>
<tr>
<td>Phone</td>
</tr>
<tr>
<td><strong>Postal Addresses</strong></td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td><strong>Street</strong></td>
</tr>
<tr>
<td><strong>City</strong></td>
</tr>
<tr>
<td><strong>State / Province</strong></td>
</tr>
<tr>
<td><strong>Postal Code / ZIP</strong></td>
</tr>
<tr>
<td><strong>Country</strong></td>
</tr>
<tr>
<td><strong>Phone</strong></td>
</tr>
</tbody>
</table>
# Postal Addresses

<table>
<thead>
<tr>
<th>Street</th>
<th>815 Boylston St</th>
<th>Street abbreviated</th>
</tr>
</thead>
<tbody>
<tr>
<td>City</td>
<td>Boston</td>
<td></td>
</tr>
<tr>
<td>State / Province</td>
<td>Massachusetts</td>
<td>State misspelled</td>
</tr>
<tr>
<td>Postal Code / ZIP</td>
<td>02116-2601</td>
<td>ZIP+4</td>
</tr>
<tr>
<td>Country</td>
<td>United States</td>
<td></td>
</tr>
<tr>
<td>Phone</td>
<td>+1 617-385-9400</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
<td>Notes</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Street</td>
<td>815 Boylston St</td>
<td>Street abbreviated</td>
</tr>
<tr>
<td>City</td>
<td>Boston</td>
<td></td>
</tr>
<tr>
<td>State / Province</td>
<td>Massachusetts</td>
<td>State misspelled</td>
</tr>
<tr>
<td>Postal Code / ZIP</td>
<td>02116-2601</td>
<td>ZIP+4</td>
</tr>
<tr>
<td>Country</td>
<td>United States</td>
<td>Localized country name</td>
</tr>
<tr>
<td>Phone</td>
<td>+1 617-385-9400</td>
<td></td>
</tr>
<tr>
<td></td>
<td>815 Boylston St</td>
<td>Street abbreviated</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>City</td>
<td>Boston</td>
<td></td>
</tr>
<tr>
<td>State / Province</td>
<td>Massachusetts</td>
<td>State misspelled</td>
</tr>
<tr>
<td>Postal Code / ZIP</td>
<td>02116-2601</td>
<td>ZIP+4</td>
</tr>
<tr>
<td>Country</td>
<td>États-Unis</td>
<td>Localized country name</td>
</tr>
<tr>
<td>Phone</td>
<td>+1 617-385-9400</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
<td>Notes</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>Street</td>
<td>815 Boylston St</td>
<td>Street abbreviated</td>
</tr>
<tr>
<td>City</td>
<td>Boston</td>
<td></td>
</tr>
<tr>
<td>State / Province</td>
<td>Massachusetts</td>
<td>State misspelled</td>
</tr>
<tr>
<td>Postal Code / ZIP</td>
<td>02116-2601</td>
<td>ZIP+4</td>
</tr>
<tr>
<td>Country</td>
<td>États-Unis</td>
<td>Localized country name</td>
</tr>
<tr>
<td>Phone</td>
<td>+1 617-385-9400</td>
<td>Dashes</td>
</tr>
</tbody>
</table>
Handling User Data Successfully

Be prepared for fuzzy input

Don’t force the user to comply with your business logic
Global Addresses

New API to support worldwide addresses
Request phonetic in addition to regular name
Native Contacts support for supplementary sub-localities (China)
When Data Goes Wrong
When Data Goes Wrong

Missing phone-number digits
When Data Goes Wrong

Missing phone-number digits

Postal code and address don’t match
When Data Goes Wrong

- Missing phone-number digits
- Postal code and address don’t match
- Misspelled email address
When Data Goes Wrong

- Missing phone-number digits
- Postal code and address don’t match
- Misspelled email address
- Otherwise unusable
Custom Errors
Custom Errors

Gracefully handle invalid or incorrect data directly in Apple Pay
Custom Errors

Gracefully handle invalid or incorrect data directly in Apple Pay

Display custom error messages
Custom Errors

Gracefully handle invalid or incorrect data directly in Apple Pay

Display custom error messages

Direct users to the specific fields that need correction
WWDC Toast
Avocado Toast Subscription
$29.99/week

Enjoy bespoke, artisanal, and delicious avocado toast every day of the week, with our weekly subscription package.

Pay with Apple Pay

CHASE SAPPHIRE...ERVE (**** 1144)
165 DAVIS STREET HILLSBOROUGH...

SHIPPING KATE BELL
SHIPPING ADDRESS INCORRECT

METHOD FREE SHIPPING
5-6 DAY DELIVERY

CONTACT KATE-BELL@MAC.COM
(555) 564-8583

PAY WWDC TOAST INC
$20.99

Shipping Address Incorrect
WWDCToast
Avocado Toast Subscription
$29.99/week

Enjoy bespoke, artisanal, and delicious avocado toast every day of the week, with our weekly subscription package.

Buy with Apple Pay

Shipping Address

SELECT A SHIPPING ADDRESS

Kate Bell’s Work
Your ZIP code doesn’t match your city

Add Shipping Address...
WWDC Toast
Avocado Toast Subscription
$29.99/week

Enjoy bespoke, artisanal, and delicious avocado toast every day of the week, with our weekly subscription package.

[Apple Pay]

Shipping Address

Kate Bell's Work
Your ZIP code doesn't match your city

Add Shipping Address...
Error Handling

Deprecated enum-based callback

```swift
public func paymentAuthorizationController(_ controller: PKPaymentAuthorizationController,
    didAuthorizePayment payment: PKPayment,
    completion: (PKPaymentAuthorizationStatus) -> Void)
```

New callback

```swift
public func paymentAuthorizationController(_ controller: PKPaymentAuthorizationController,
    didAuthorizePayment payment: PKPayment,
    handler completion: (PKPaymentAuthorizationResult) -> Void)
```
Error Handling

Deprecated enum-based callback

```swift
public func paymentAuthorizationController(_ controller: PKPaymentAuthorizationController, didAuthorizePayment payment: PKPayment, completion: (PKPaymentAuthorizationStatus) -> Void)
```

New callback

```swift
public func paymentAuthorizationController(_ controller: PKPaymentAuthorizationController, didAuthorizePayment payment: PKPayment, handler completion: (PKPaymentAuthorizationResult) -> Void)
```
PKPaymentAuthorizationResult

Contains a status and an array of errors

Errors are standard `NSError`s describing what went wrong

```swift
open class PKPaymentAuthorizationResult : NSObject {
    public init(status: PKPaymentAuthorizationStatus, errors: [Error])

    open var status: PKPaymentAuthorizationStatus
    open var errors: [Error]
}
```
PKPaymentAuthorizationResult

Errors

Each error represents an individual problem with the payment

Use `NSError` codes to indicate the issue

Use User Info Keys to provide specific resolution steps and information

```swift
open class PKPaymentAuthorizationResult : NSObject {
    public init(status: PKPaymentAuthorizationStatus, errors: [Error])

    open var status: PKPaymentAuthorizationStatus
    open var errors: [Error]
}
```
// Swift
let countryError = NSError.paymentShippingAddressInvalidError(withKey: CNPostalAddressCountryKey, localizedDescription: "Only US delivery supported")

// ES6
const countryError = new ApplePayError("shippingContactInvalid", "country", "Only US delivery supported");
Error Handling
New callbacks

New callbacks for most existing delegate methods

New objects allow for extensible messaging

```swift
func paymentAuthorizationController(_ controller: PKPaymentAuthorizationController,
   didSelectShippingContact contact: PKContact,
   handler completion: (PKPaymentRequestShippingContactUpdate) -> Void)
```
Error Handling
On the web

Like-for-like API is available in Safari 11

Existing completion functions have been overloaded

// Old API
session.completePaymentMethodSelection(update['newTotal'], update['newLineItems']);
session.completePayment(update['status']);
Error Handling
On the web

Like-for-like API is available in Safari 11

Existing completion functions have been overloaded

// Old API
session.completePaymentMethodSelection(update['newTotal'], update['newLineItems']);
session.completePayment(update['status']);

// New API
session.completePaymentMethodSelection(update);
session.completePayment(update);
Demo

Jon Nguy, Software Engineer
What’s New with Wallet
What’s New with Wallet
A lick of paint

New mass deletion and management interface

Lighter color scheme

Good time to revisit your pass designs
What’s New with Wallet
NFC passes

NFC passes let you send customer information over NFC

Only encrypted NFC passes supported from iOS 11

Register for NFC passes at developer.apple.com/apple-pay
What’s New with Wallet

Sharing

Passes can now be opted out of sharing

Useful for single use items like loyalty cards or tickets
What’s New with Wallet

Sharing

Sharing disablement is a convenience (not a security) feature

```
{
    "teamIdentifier" : "K123456",
    "passTypeIdentifier" : "pass.com.apple.store",
    "organizationName" : "Apple Store",
    "serialNumber" : "123456789",
    "description" : "Apple Store Gift Card",
    "sharingProhibited" : true,
    ...
}
```
Summary

Guidelines for building best-in-brand Apple Pay experiences

New API for handling complex errors

New ways to share and interact with passes and NFC
More Information

### Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>WWDC 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple Pay on the Web</td>
<td></td>
</tr>
<tr>
<td>What’s New with Wallet and Apple Pay</td>
<td>WWDC 2016</td>
</tr>
<tr>
<td>Introducing Core NFC</td>
<td>WWDC 2017 Video</td>
</tr>
</tbody>
</table>
Apple Pay and Wallet Lab | Technology Lab J | Fri 9:00AM–12:00PM