

[ProximityReader](#) / [PaymentCardReaderSession](#)

Class

PaymentCardReaderSession

The object you use to start reading a contactless payment or loyalty card.

iOS 15.4+ | iPadOS 15.4+ | Mac Catalyst 17.0+

```
class PaymentCardReaderSession
```

Mentioned in

- 📄 Adding support for Tap to Pay on iPhone to your app
- 📄 Accepting loyalty passes from Wallet

Overview

Use a `PaymentCardReaderSession` object to read payment and loyalty cards from a properly configured device. You don't create this object directly. Instead, you obtain one by calling the `prepare(using:)` method of your `PaymentCardReader` object, which returns a session after the successful configuration of the device.

Maintain a strong reference to a session object for the duration of the card-reading process. You may use the same session object to perform multiple read operations, but you may perform only one read operation at a time from the device.

Topics

[Reading a payment card](#)

```
func readPaymentCard(PaymentCardTransactionRequest) async throws -> PaymentCardReadResult
```

Presents a sheet to read a contactless payment card for a purchase or a refund, and returns the encrypted card data.

```
func readPaymentCard(PaymentCardVerificationRequest) async throws -> PaymentCardReadResult
```

Presents a sheet to verify a contactless payment card, and returns the card data.

Reading a loyalty card

```
func readPaymentCard(PaymentCardTransactionRequest, vasRequest: VASRequest, stopOnVASResult: Bool) async throws -> (PaymentCardReadResult?, VASReadResult?)
```

Presents a sheet to read both contactless payments and loyalty cards for a purchase or refund, and returns the relevant card data.

```
func readVAS(VASRequest) async throws -> VASReadResult
```

Presents a sheet to read a loyalty card for Value Added Services (VAS), and returns the loyalty card data.

Requesting the PIN

```
func capturePIN(using: PaymentCardReaderSession.PINToken, cardReader TransactionID: String) async throws -> PaymentCardReadResult
```

Presents a sheet to capture the PIN when required by the payment card issuer, and returns the previously encrypted card data including newly captured PIN data.

```
struct PINToken
```

A secure PIN token that you receive from your participating payment service provider.

Cancelling the reading process

```
func cancelRead() async throws -> Bool
```

Dismiss the sheet that prompts someone to present their card for reading.

Getting error information

```
enum ReadError
```

Errors that can occur during a card read.

Deprecated

```
func readPaymentCard(PaymentCardTransactionRequest, eventHandler: ((PaymentCardReaderSession.Event) -> Void)?) async throws -> PaymentCardReadResult
```

Presents a sheet to read a contactless payment card for a purchase or a refund, and returns the encrypted card data.

Deprecated

```
func readPaymentCard(PaymentCardVerificationRequest, eventHandler: ((PaymentCardReaderSession.Event) -> Void)?) async throws -> PaymentCardReadResult
```

Presents a sheet to verify a contactless payment card, and returns the card data.

Deprecated

```
func readPaymentCard(PaymentCardTransactionRequest, vasRequest: VASRequest, stopOnVASResult: Bool, eventHandler: ((PaymentCardReaderSession.Event) -> Void)?) async throws -> (PaymentCardReadResult?, VASReadResult?)
```

Presents a sheet to read both contactless payments and loyalty cards for a purchase or refund, and returns the relevant card data.

Deprecated

```
func readVAS(VASRequest, eventHandler: ((PaymentCardReaderSession.Event) -> Void)?) async throws -> VASReadResult
```

Presents a sheet to read a loyalty card for Value Added Services (VAS), and returns the loyalty card data.

Deprecated

```
let id: String
```

A unique identifier for this object.

Deprecated

```
enum Event
```

Optional events you can observe during the card-reading process.

Deprecated

Instance Properties

```
let currentOSVersionDeprecationDate: Date?
```

The date when current OS version will be deprecated.

Relationships

Inherited By

`StoreAndForwardPaymentCardReaderSession`

Conforms To

`Sendable`, `SendableMetatype`

See Also

Payment card reader

 [Setting up Tap to Pay on iPhone](#)

Request and configure the required entitlement to support Tap to Pay on iPhone.

 [Adding support for Tap to Pay on iPhone to your app](#)

Configure your app to use Tap to Pay on iPhone to read contactless payment cards.

`class PaymentCardReader`

An object you use to configure Tap to Pay on iPhone on the current device.