

[ARKit](#) / ARKitSession

## Class

# ARKitSession

The main entry point for receiving data from ARKit.

macOS 26.0+ | visionOS 1.0+

```
final class ARKitSession
```

## Overview

Sessions in ARKit require either implicit or explicit authorization. To explicitly ask for permission for a particular kind of data and choose when a person is prompted for that permission, call [requestAuthorization\(for:\)](#) before [run\(\\_:\)](#).

The following shows a session that starts by requesting implicit authorization to use world sensing:

```
let planeData = PlaneDetectionProvider(alignments: [.horizontal, .vertical])

Task {
    do {
        try await self.session.run([planeData])
        // Update app based on the planeData.anchorUpdates async sequence.
    } catch {
        print("ARKitSession error:", error)
    }
}
```

Because a [PlaneDetectionProvider](#) instance's required authorizations include [ARKitSession.AuthorizationType.worldSensing](#), the system asks someone using your app to permit world sensing before ARKit supplies any of that kind of data.

### Note

ARKit stops sessions when they're deinitialized; keep a reference to a session instance for as long as the session needs to run.

## Topics

### Starting and stopping a session

`convenience init()`

Creates a new session.

`func run([any DataProvider]) async throws`

Runs a session with the data providers you supply.

`func stop()`

Stops all data providers running in this session.

`struct Error`

An error that might occur when running data providers on an ARKit session.

### Getting authorization

`func requestAuthorization(for: [ARKitSession.AuthorizationType]) async  
-> [ARKitSession.AuthorizationType : ARKitSession.AuthorizationStatus]`

Requests authorization from the user to use the specified kinds of ARKit data.

`enum AuthorizationType`

The authorization types you can request from ARKit.

`func queryAuthorization(for: [ARKitSession.AuthorizationType]) async ->  
[ARKitSession.AuthorizationType : ARKitSession.AuthorizationStatus]`

Checks whether the current session is authorized for particular authorization types without requesting authorization.

`enum AuthorizationStatus`

The authorization states for a type of ARKit data.

### Observing a session

`var events: ARKitSession.Events`

An asynchronous sequence of events that provide updates to the current authorization status of the session.

`struct Events`

A sequence of events.

`enum Event`

Enumeration of possible session events.

`var description: String`

A textual representation of this session.

## Initializers

`convenience init(device: RemoteDeviceIdentifier)`

Create a new session connected to the specified device.

## Instance Properties

`var dataProviders: [any DataProvider]`

A list of all data providers on this session.

---

# Relationships

## Conforms To

CustomStringConvertible, Sendable, SendableMetatype

---

# See Also

## visionOS



Setting up access to ARKit data

Check whether your app can use ARKit and respect people’s privacy.

`protocol` `DataProvider`

A source of live data from ARKit.

`protocol` `Anchor`

The identity, location, and orientation of an object in world space.

☰ `ARKit in visionOS`  
Create immersive augmented reality experiences.