

[TabletopKit](#) / TableSeat

Protocol

TableSeat

A protocol for seats at the table that players occupy.

visionOS 2.0+

```
protocol TableSeat : Identifiable where Self.ID == TableSeatIdentifier
```

Overview

To represent seats in your game, follow these steps:

1. Create a structure that conforms to this protocol.
2. Set the State type alias to TableSeatState.
3. Declare the id property as a TableSeatIdentifier structure.
4. Declare the initialState property as a State structure.
5. Implement an initializer that sets these properties.

```
struct Seat: TableSeat {
    typealias State = TableSeatState
    var id: TableSeatIdentifier
    var initialState: State

    init(index: Int = 0, position: TableVisualState.Point2D) {
        id = .seat(index)
        initialState = .init(pose: .init(position: position, rotation: .init()))
    }
}
```

Then add the seats to the `TableSetup` object using one of its add methods. For example, use the `add(seat:)` method to specify a position for the seat.

```
var setup = TableSetup(tabletop: table)
setup.add(seat: Seat(index: 0, position: .init(x: 0, z: -0.5)))
```

To render seats using RealityKit, conform to the `EntityTableSeat` protocol instead.

Topics

Getting the seat data

```
var initialState: Self.State
    Required

associatedtype State : SeatState
    Required
```

Relationships

Inherits From

Identifiable

Inherited By

EntityTableSeat

See Also

Seats

```
struct TableState
    The state of the table that can be queried and modified.
```

```
protocol EntityTableSeat
```

A protocol for seats at the table that you render using RealityKit.

```
struct TableSeatIdentifier
```

A unique identifier for seats.

```
struct TableSeatState
```

The data associated with a seat that a player occupies.

```
protocol SeatState
```

A protocol for seat data that TabletopKit syncs between players.