

[MapKit](#) / MapKit for SwiftUI

API Collection

MapKit for SwiftUI

MapKit for SwiftUI allows you to build map-centric views and apps across Apple platforms. You can design expressive and highly interactive Maps with minimal code by composing views, using ViewBuilders and view modifiers.


Overview

Like MapKit for AppKit and UIKit, MapKit for SwiftUI allows you to take advantage of map styles ranging from satellite imagery to rich, 3D perspective imagery to present vivid maps. Using [MapContentBuilder](#) you can configure your maps to show [Marker](#) and [Annotation](#) views, or — for more specialized content — you can design your own SwiftUI views to place on the map. To add even more interactivity, MapKit for SwiftUI supports overlays to highlight areas on the map, enabling you to animate paths and directions using [MapPolyline](#), or make it easy for people to dig deeper into on the ground details with tappable points of interest. People who use your app can also explore at street level using [LookAroundPreview](#) and Look Around viewer.

Note

For more information about integrating MapKit into your app using SwiftUI, see WWDC23 session 10043: [Meet MapKit for SwiftUI](#)

Topics

 Searching, displaying, and navigating to places

Convert place information between coordinates and user-friendly place names, get cycling directions, and conveniently display formatted addresses.

Essentials

`struct Map`

A view that displays an embedded map interface.

`struct MapStyle`

A style that you can apply to a map.

Annotations and overlays

`struct Annotation`

A customizable annotation used to indicate a location on a map.

`struct MapCircle`

A circular overlay with a configurable radius that you center on a geographic coordinate.

`struct MapPolygon`

A closed polygon overlay.

`struct MapPolyline`

An open polygon overlay consisting of one or more connected line segments.

`struct Marker`

A balloon-shaped annotation that marks a map location.

`struct UserAnnotation`

Displays the person's current location on the map.

Map controls

`struct MapCompass`

A view that reflects the current orientation of the associated map.

`struct MapLocationCompass`

A view that displays a combined user location button and map compass.

`struct MapPitchSlider`

A slider control that allows a person to change the pitch of the map.

`struct MapPitchToggle`

A button that sets the pitch of the associated map.

`struct MapScaleView`

Displays a legend with distance information for the associated map.

`struct MapUserLocationButton`

A button that sets the framing of the associated map to the user location.

`struct MapZoomStepper`

Buttons a person uses to adjust the zoom level of the map.

Exploring at street level

`struct LookAroundPreview`

A view that provides a Look Around preview for a specific geographic location.

Map features

`struct MapFeature`

A tappable map feature.

`struct MapSelection`

A value representing a selected feature on a map.

`protocol MapSelectable`

Map customization

`struct MapCamera`

Defines a virtual viewpoint above the map surface.

`struct MapCameraBounds`

Defines an optional boundary of an area within which the map's center needs to remain.

`struct MapCameraPosition`

A structure that describes how to position the map's camera within the map.

`struct MapCameraUpdateContext`

A structure that defines additional information about the map camera.

`struct MapCameraUpdateFrequency`

A structure that describes when the map camera updates.

Place information

```
struct MapItemDetailSelectionAccessoryStyle
```

The map item detail selection accessory style.

```
func mapItemDetailSelectionAccessory(MapItemDetailSelectionAccessoryStyle?) -> some MapContent
```

Specifies the selection accessory to display for the selected map item content.

```
static func callout(MapItemDetailSelectionAccessoryStyle.CalloutStyle) -> MapItemDetailSelectionAccessoryStyle
```

Presents the accessory as an annotation callout on the map.

Geocoding

```
class MKGeocodingRequest
```

A class that looks up a geographic coordinate using the provided string.

```
class MKReverseGeocodingRequest
```

A class that looks up address strings for the provided geographic coordinates.

Representing places and addresses

```
class MKMapItem
```

A point of interest on the map.

```
class MKAddress
```

A class that contains a full address, and, optionally, a short address.

```
class MKAddressRepresentations
```

A class that provides formatted address strings.



GeoToolbox

Determine place descriptor information for map coordinates.

Points of interest

```
struct PointOfInterestCategories
```

A structure you use to define points of interest to include or exclude on a map.

Protocols

`protocol DynamicMapContent`

A type of view that generates views from an underlying collection of data.

`protocol MapContent`

A protocol used to construct map content such as controls, markers, and annotations.

`struct MapContentBuilder`

A result builder that creates map content from closures you provide.

`struct MapContentView`

A view that contains content that displays on a map at a specific position, and that responds to specific interactions you specify.

Structures

`struct DefaultUserAnnotationContent`

A structure that represents the view to show at the user's location on the map.

`struct EmptyMapContent`

A map content element that doesn't contain any content.

`struct MapProxy`

A proxy for accessing sizing information about a given map view.

`struct MapReader`

A container view that defines its contents as a function of information about the first contained map.

`struct TupleMapContent`

A view created from a Swift tuple of map content values.

`struct MapSelectableContentView`

See Also

The MapKit APIs

☰ MapKit for AppKit and UIKit

📄 Adopting unified Maps URLs

Access Maps URLs and options for displaying Maps information across Apple platforms.