Sample Code

Optimizing Map Views with Filtering and Camera Constraints

Display a map that is relevant to the user by filtering points of interest and search results, and constraining the visible region.

Download

iOS 13.0+ | iPadOS 13.0+ | Xcode 11.0+

Overview

This sample code demonstrates three features that can make a map relevant to your users:

- Constraining the map view's visible region and zoom to keep the desired areas in view.
- Filtering the points of interest to reduce the clutter on the map view.
- Filtering search result types and points of interest to improve to search and autocompletion.

> **Note**
>
> This sample code project is associated with WWDC 2019 session 236: What's New in MapKit and MapKit JS.

See Also

Points of interest

Identifying unique locations with Place IDs

Obtain information about a point of interest that persists over its lifetime.

`class` `MKMapFeatureAnnotation`

A class that describes an annotation element on the map's display such as a point of interest, territorial boundary, or physical feature.

`struct` `MKMapFeatureOptions`

A structure you use to tell the map which kinds of features users can interact with.

`class` `MKMapItemRequest`

A utility class you use to request additional information about a map feature.

`class` `MKIconStyle`

A class you use to customize the annotation view icon of a point of interest (POI) on the map.

`class` `MKPointOfInterestFilter`

A filter that includes or excludes point of interest categories from a map view, local search, or local search completer.

`struct` `MKPointOfInterestCategory`

A point of interest category.