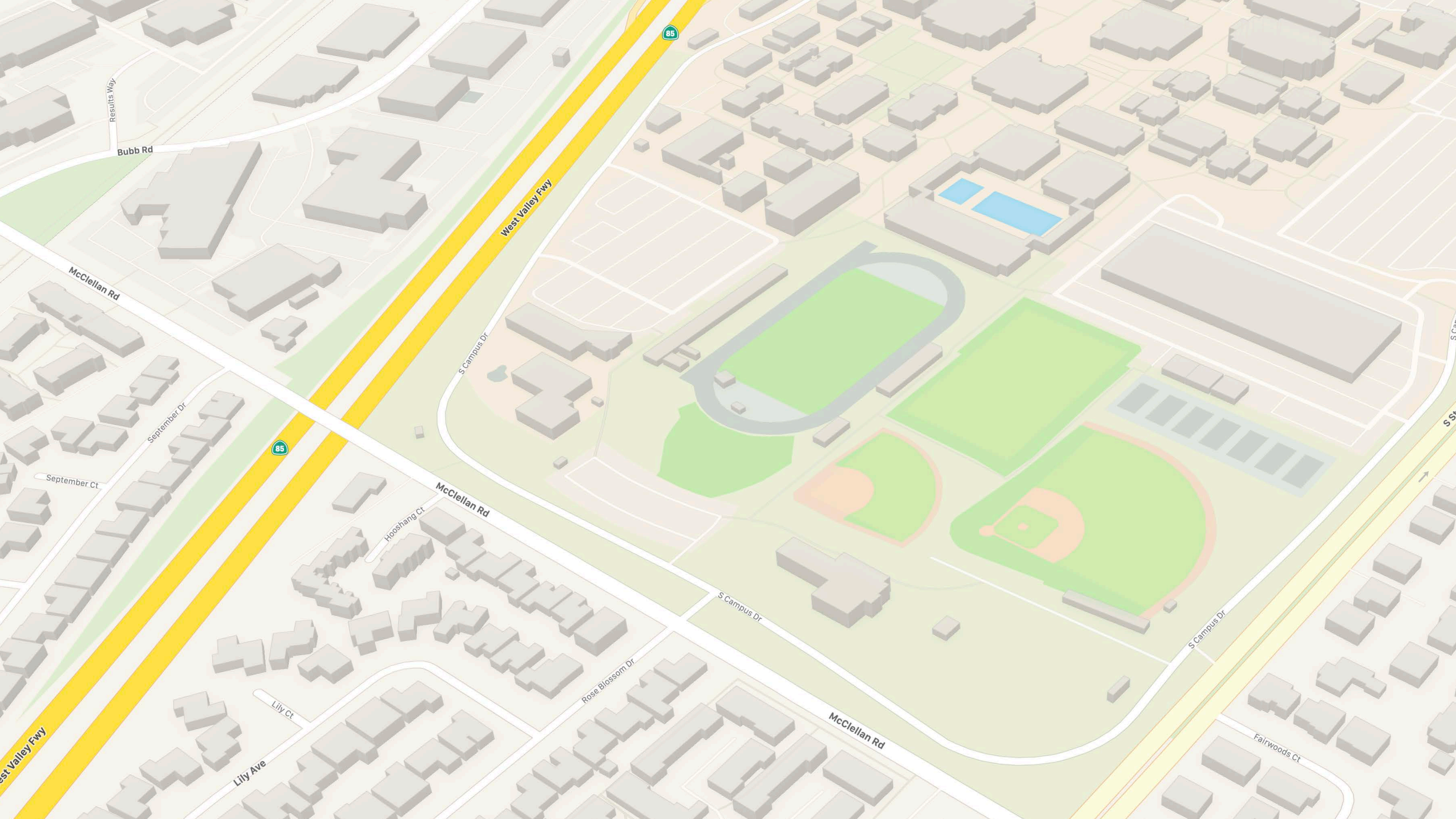


#WWDC19

What's New in MapKit and MapKit JS

Alexander Jakobsen, MapKit Engineer

Nalini Shah, MapKit Engineer



85

85

West Valley Fwy

Bubb Rd

McClellan Rd

September Dr

September Ct

S Campus Dr

McClellan Rd

Hooshang Ct

S Campus Dr

Rose Blossom Dr

Lily Ct

Lily Ave

McClellan Rd

S Campus Dr

Fairwoods Ct



Imperial Ave

Bubb Rd

West Valley Fwy

N Campus Dr

N Campus Dr

Stevens Creek Blvd

N Campus Dr

Anton Way

Congress

Patriot Way

Alves Dr

Serale Way

Freedom Dr

United Pl

N Stelling Rd

N Stelling Rd

Bubb Rd

West Valley Fwy

S Campus Dr

September Dr

85

McClellan Rd

Hooshang Ct

S Campus Dr

Rose Blossom Dr

McClellan Rd

S Campus Dr

S Stelling Rd

S Campus Dr

Pepper Tree Ln

Bonny Dr

Lily Ave

Lily Ct

Rose Blossom Dr

Red Fir Ct

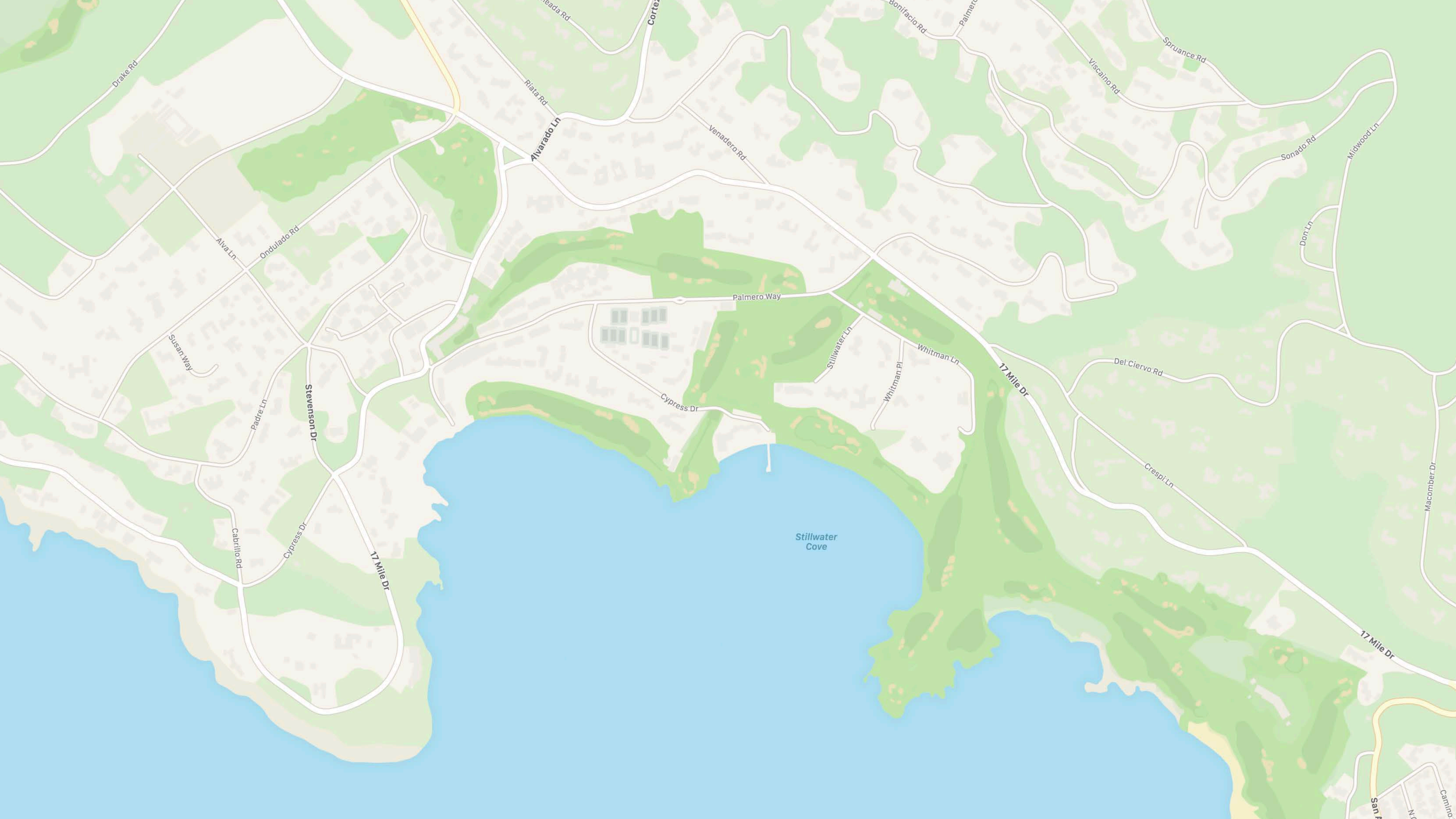
Fairwoods Ct

Tula Ln

Sola St

Tula Ct

Tula Ln



Drake Rd

Alva Ln

Ondulado Rd

Susan Way

Padre Ln

Stevenson Dr

Cabrillo Rd

Cypress Dr

17 Mile Dr

Riata Rd

Alvarado Ln

Corte

Venadero Rd

Palmero Way

Cypress Dr

Stillwater Ln

Whitman Pl

Whitman Ln

17 Mile Dr

Del Ciervo Rd

Crespi Ln

17 Mile Dr

San A

N

Camini

leada Rd

Bonifacio Rd

Palmer

Viscaino Rd

Spruance Rd

Sonado Rd

Midwood Ln

Don Ln

Macomber Dr



Signal Hill Rd
17 Mile Dr

Sunset Pt
Portola Rd
Porque Ln
17 Mile Dr
Padre Ln
Sombria Ct

Laureles Ln
Sombria Ln
Drake Rd
Padre Ln
17 Mile Dr

Founders Ln
Bristol Ln
Silver Ct
Faculty Dr
Bristol Curve
Forest Lake Rd
Lisbon Ln

Stevenson Dr
Drake Rd
Alvarado Ln
Riaia Rd
Oleada Rd

Alva Ln
Ondulado Rd
Susan Way
Padre Ln
Stevenson Dr
Cypress Dr
Cabrilho Rd
17 Mile Dr

Douglas Way
Lisbon Ln
Viscaino Rd
Ronda Rd

DEL MONTE FOREST
Cortez Rd
Bonifacio Rd
Palmero Way

Venadero Rd
Palmero Way
Stillwater Ln
Whitman Pl
Whitman Ln
Cypress Dr
17 Mile Dr

Stillwater Cove



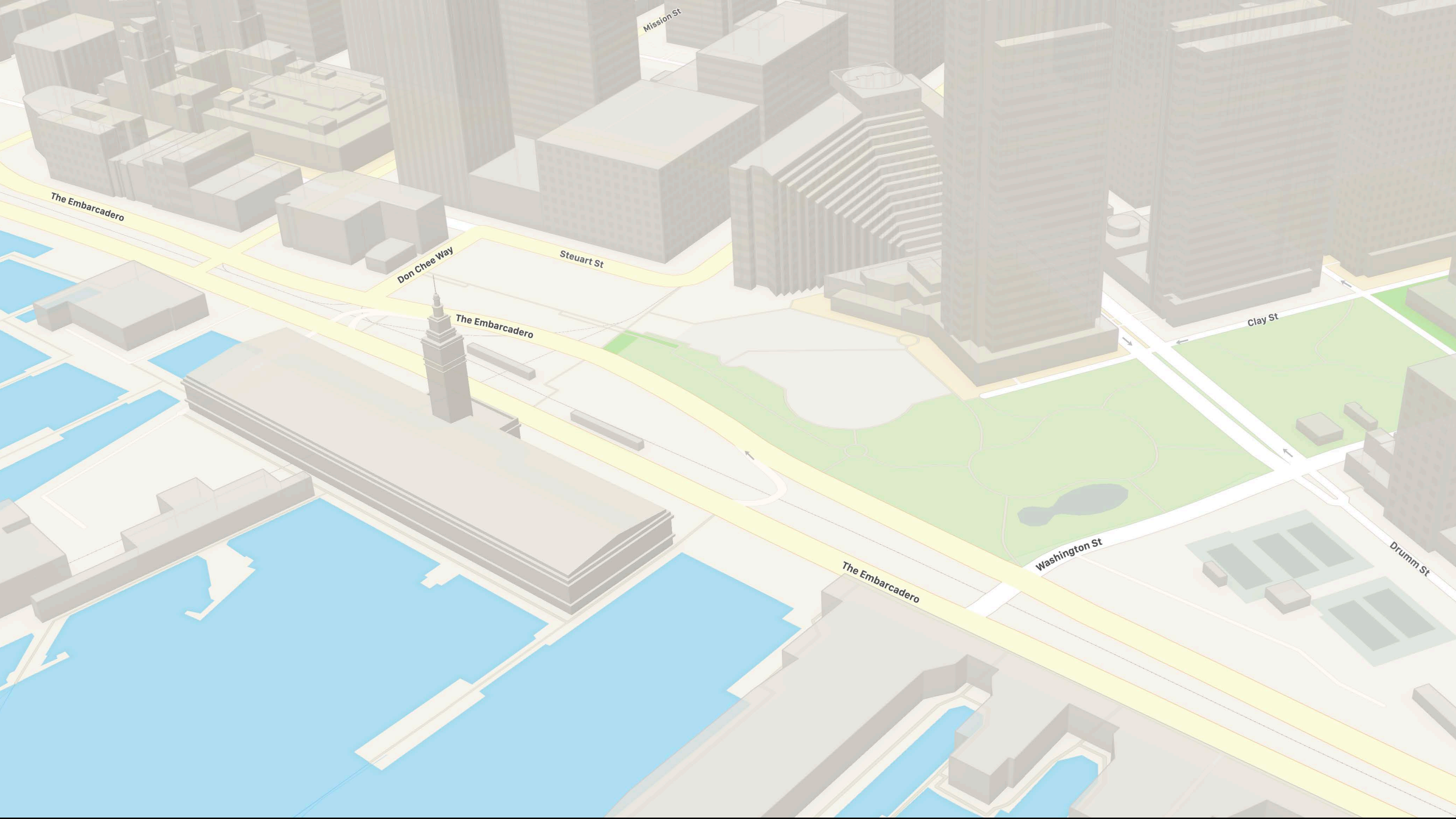
Embarcadero

Don Chee Way

Steuart St

The Embarcadero

The Embarcadero



Mission St

The Embarcadero

Don Chee Way

Steuart St

The Embarcadero

Clay St


Washington St

The Embarcadero

Drumm St

duckduckgo.com/?q=san+francisco&t=h_&ia=news&iax=about&iaxm=about

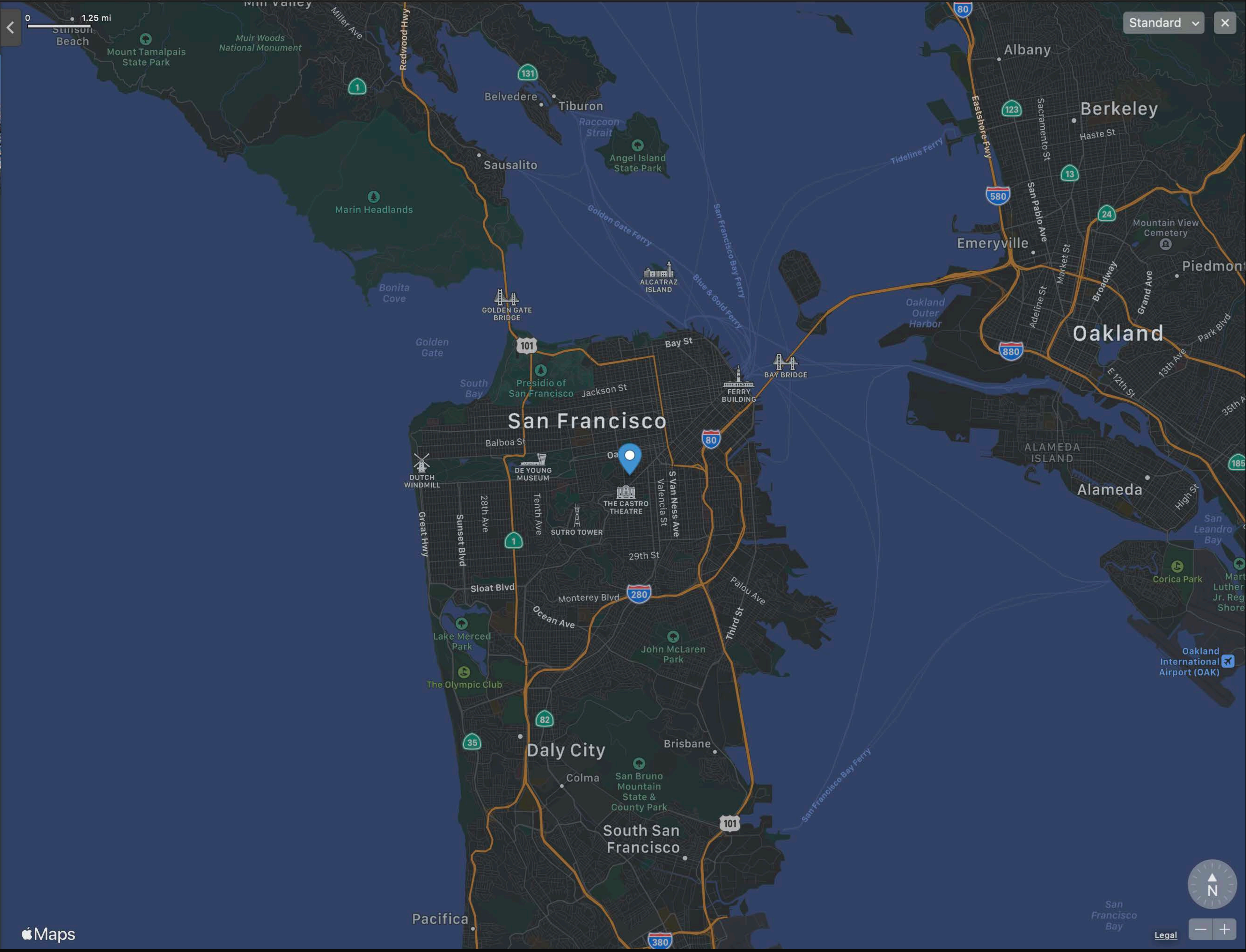
Search Results



San Francisco [Directions](#)
sf.gov

San Francisco, officially the City and County of San Francisco, is the cultural, commercial, and financial center of Northern California. It covers an area of about, mostly at the north end of the San Francisco Peninsula in the San Francisco Bay Area. San Francisco is the fourth-most populous city in California and the thirteenth-most populous in the United States, with a 2017 census-estimated population of 884,363. The consolidated city-county is also the fifth most densely populated U.S. county, behind only four of the five New York City boroughs. As of 2016, it was the 7th highest-income county in the United States, with a per capita personal income of \$110,418. [Wikipedia](#)

CSA: San Jose–San Francisco–Oakland
Metro: San Francisco–Oakland–Hayward
Mission: June 29, 1776
Incorporated: April 15, 1850
Founded by: José Joaquín Moraga, Francisco Palóu
Named for: St. Francis of Assisi
Type: Mayor-council
Body: Board of Supervisors
Mayor: London Breed (D)
Assembly members: David Chiu (D), Phil Ting (D)
State senator: Scott Wiener (D)
United States Representatives: Nancy Pelosi (D), Jackie Speier (D)



Apple Maps

MacBook Pro

9:41



WWDC Companion

Find Accommodation

Check out our partners for a great deal!



After Hours

Find the best places in San Jose



Concert in the Park

Thursday night at 8 PM. Be there!



9:41



WWDC Companion

Find Accommodation

Check out our partners for a great deal!



After Hours

Find the best places in San Jose



Concert in the Park

Thursday night at 8 PM. Be there!

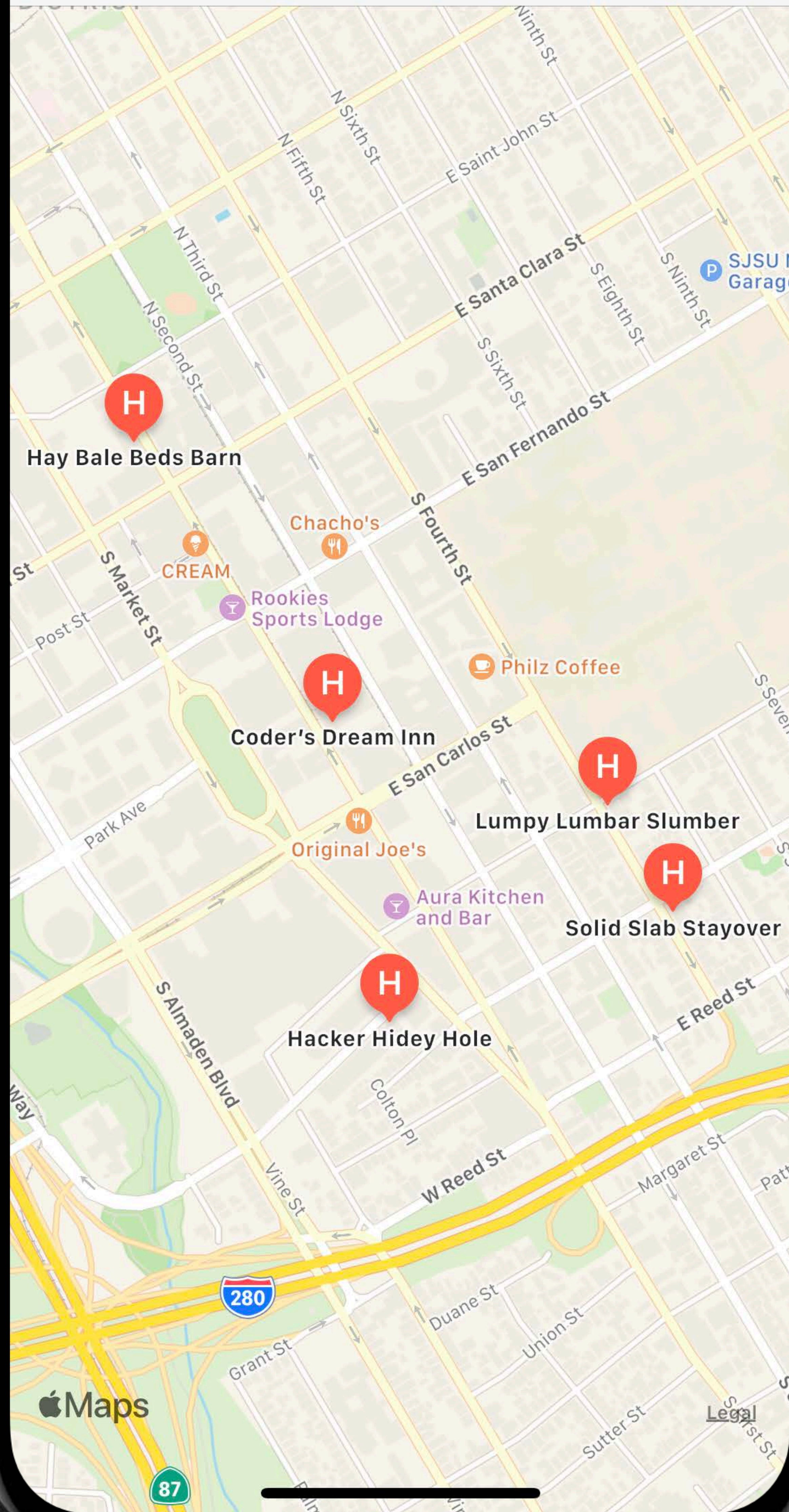


9:41



[Back](#)

Accommodation



Apple Maps

87

Legal

9:41



WWDC Companion

Find Accommodation

Check out our partners for a great deal!



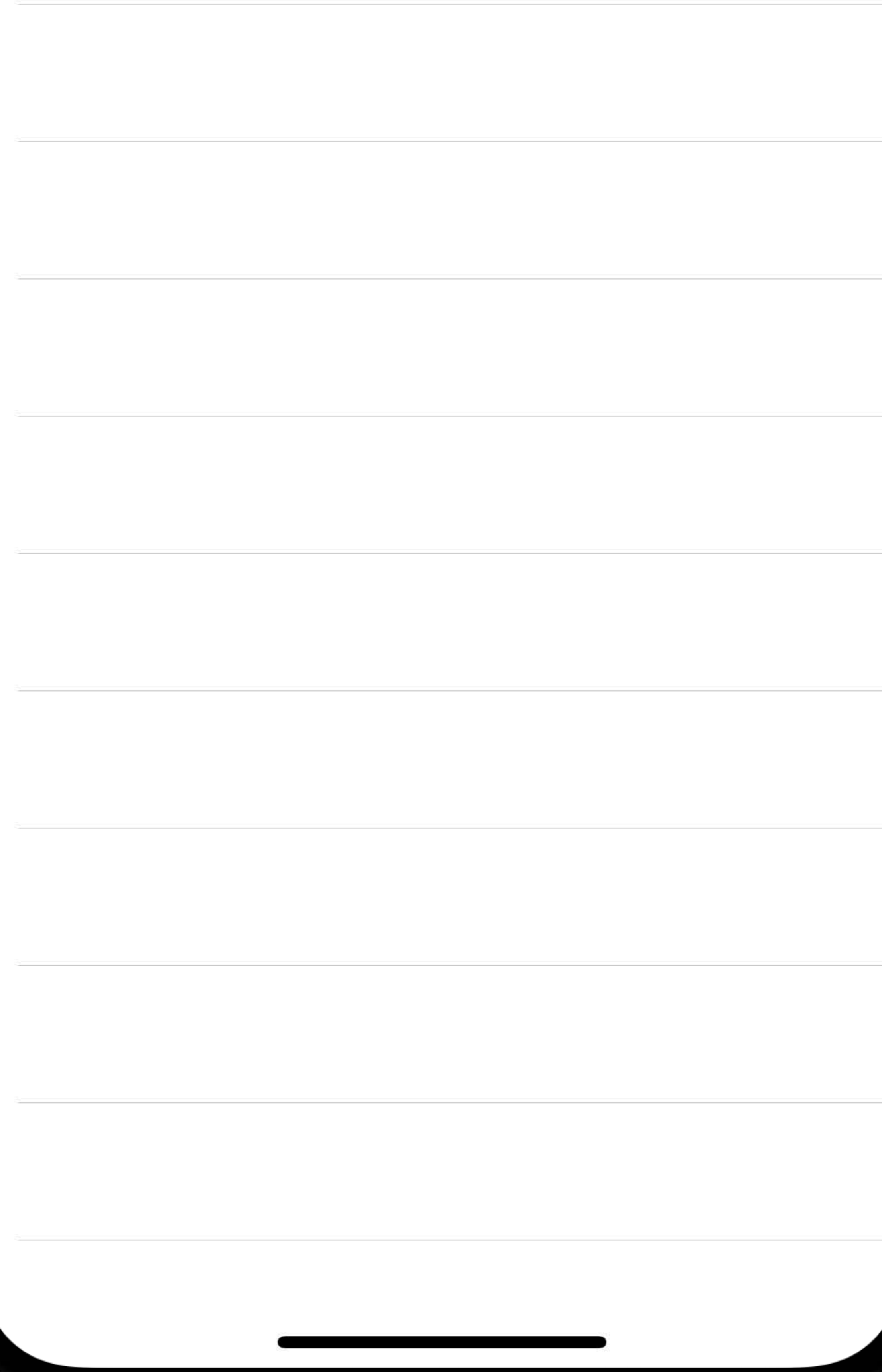
After Hours

Find the best places in San Jose



Concert in the Park

Thursday night at 8 PM. Be there!



9:41

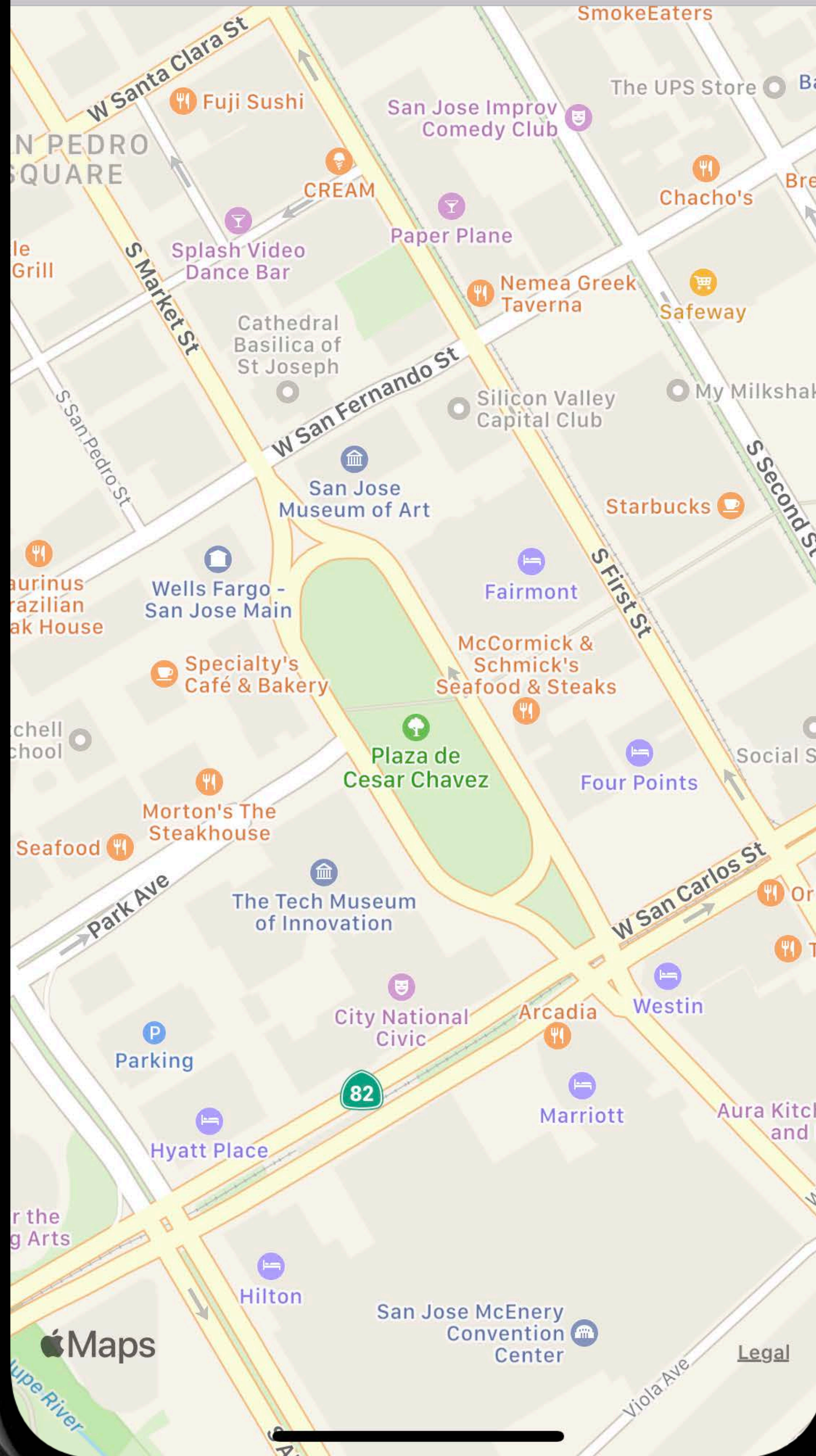


< Back

After Hours

Search for restaurants & bars

Cancel



Apple Maps

Legal

9:41



< Back

After Hours

Q Mor|



Cancel

Morton's The Steakhouse

177 Park Ave, San Jose, CA 95113, United States

q w e r t y u i o p

a s d f g h j k l

⬆ z x c v b n m ⬇

123

space

Search



9:41



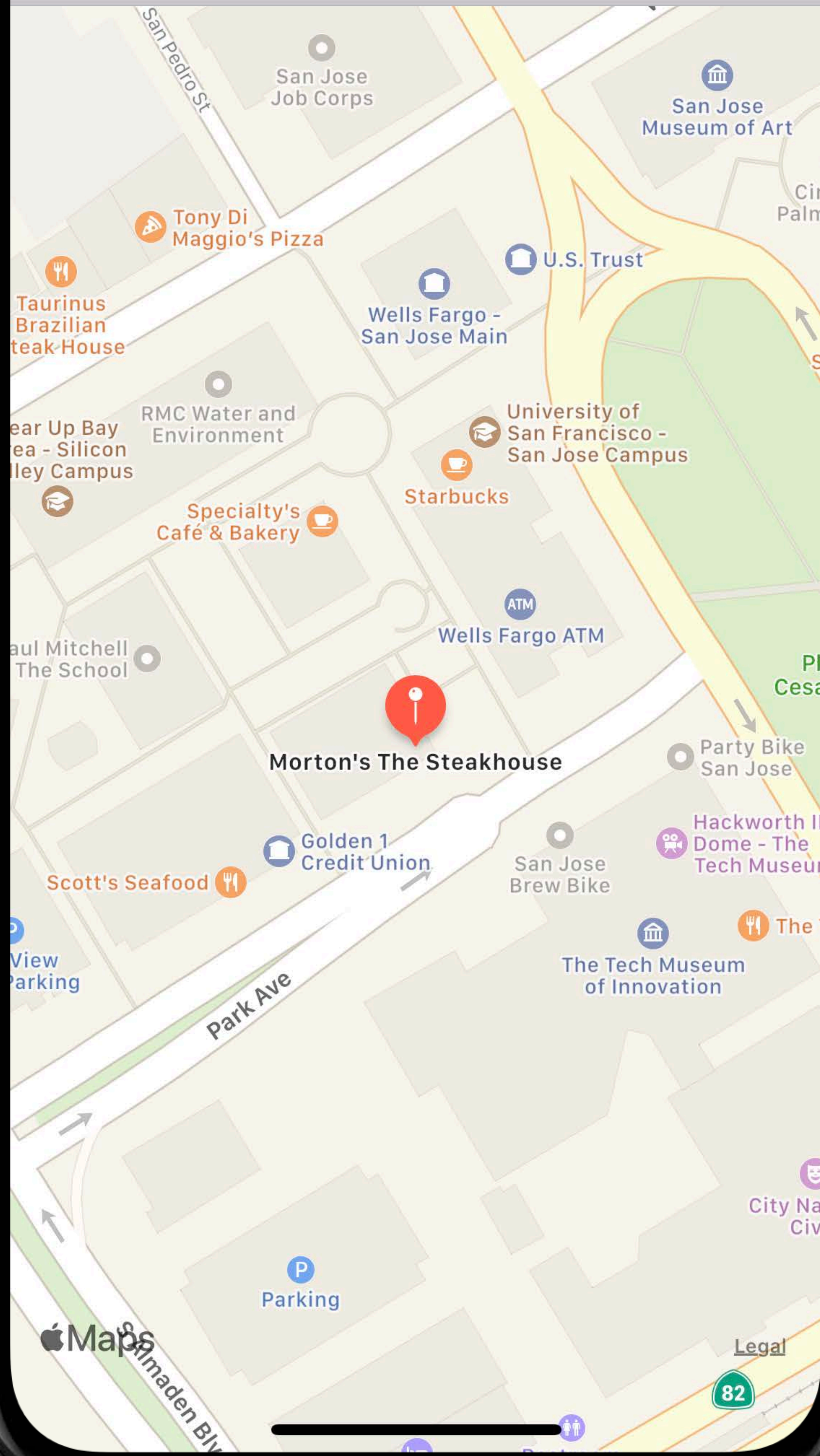
< Back

After Hours

🔍 Morton



Cancel



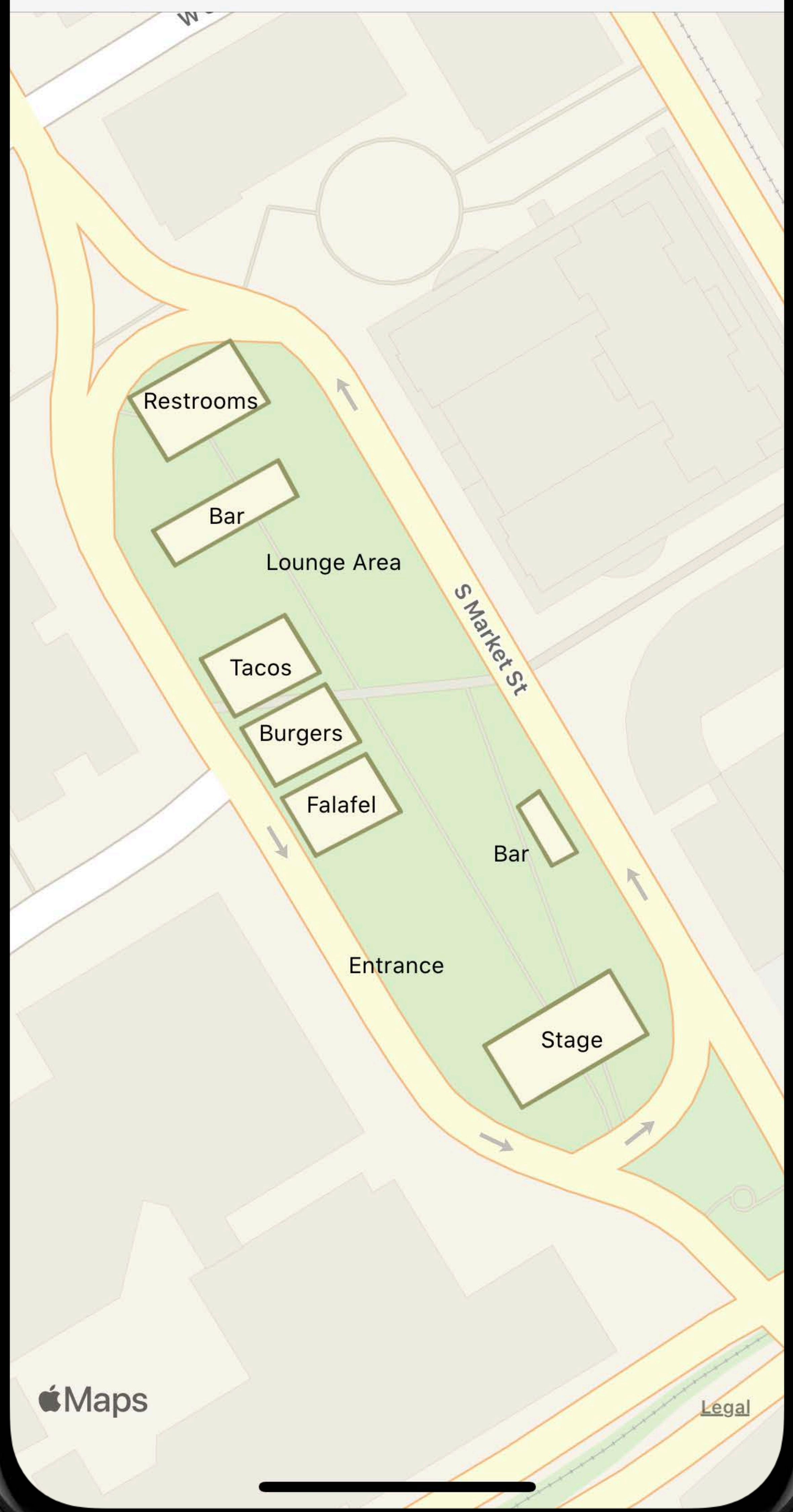
Apple Maps

9:41



[Back](#)

Concert in The Park



Apple Maps

Legal

Agenda

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

9:41



< Contacts

Edit

Steve Jobs Theater

Apple Building

Directions



Share

Address

Apple Inc.
10501 N Tantau Ave
Cupertino, CA 95014
United States

Report an Issue



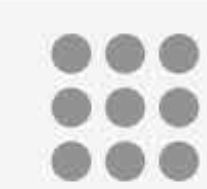
Favorites



Recents



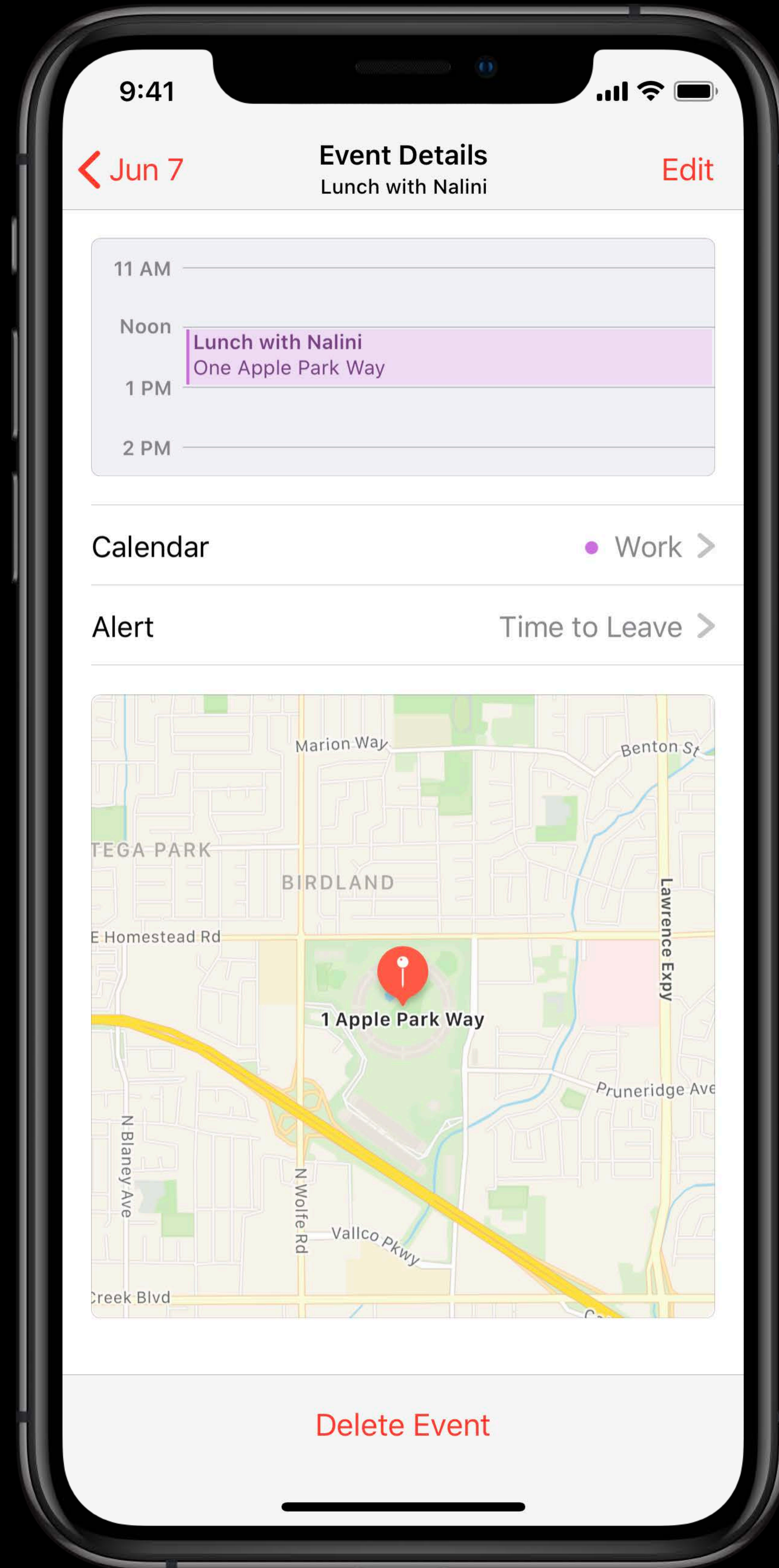
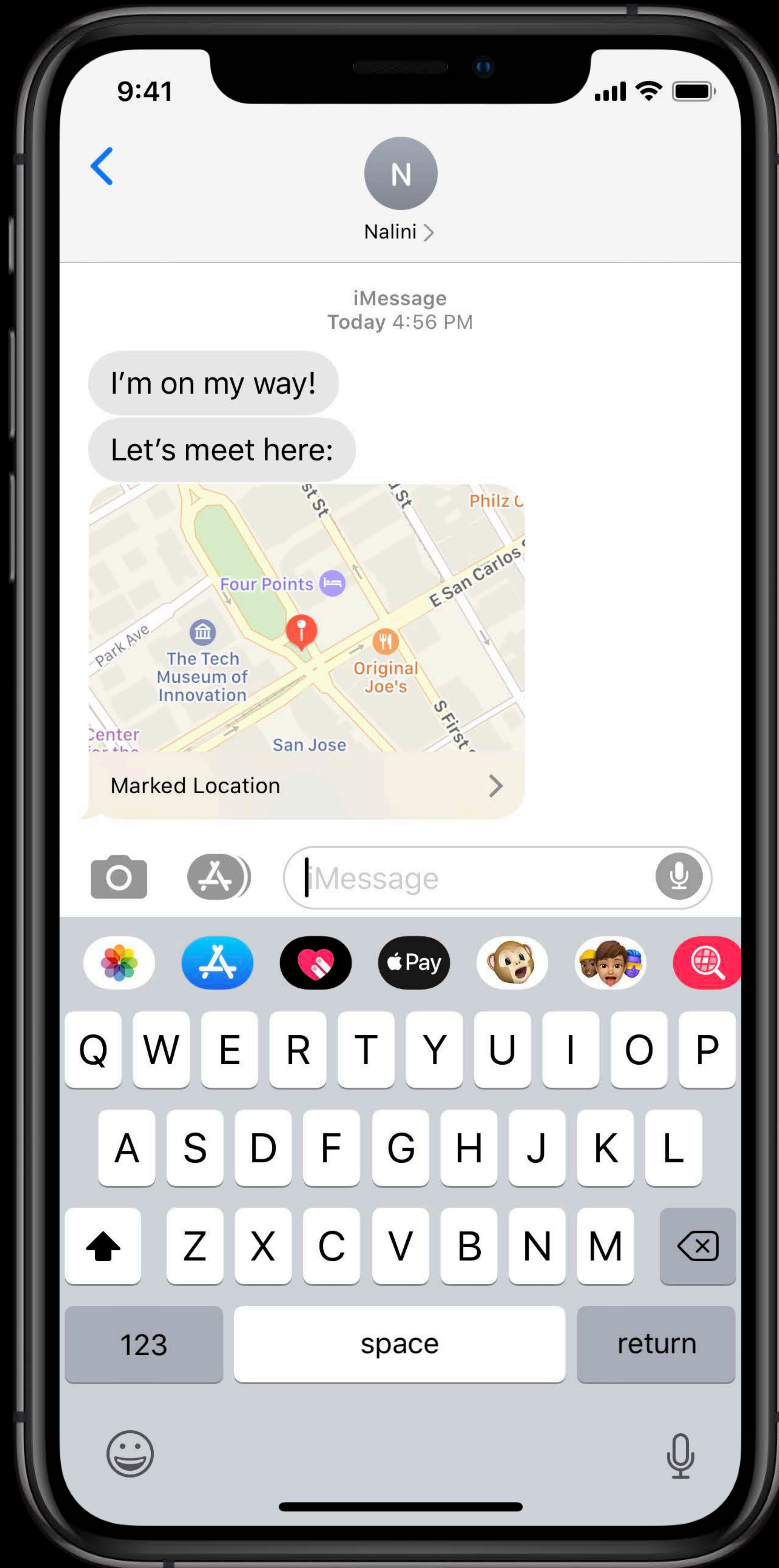
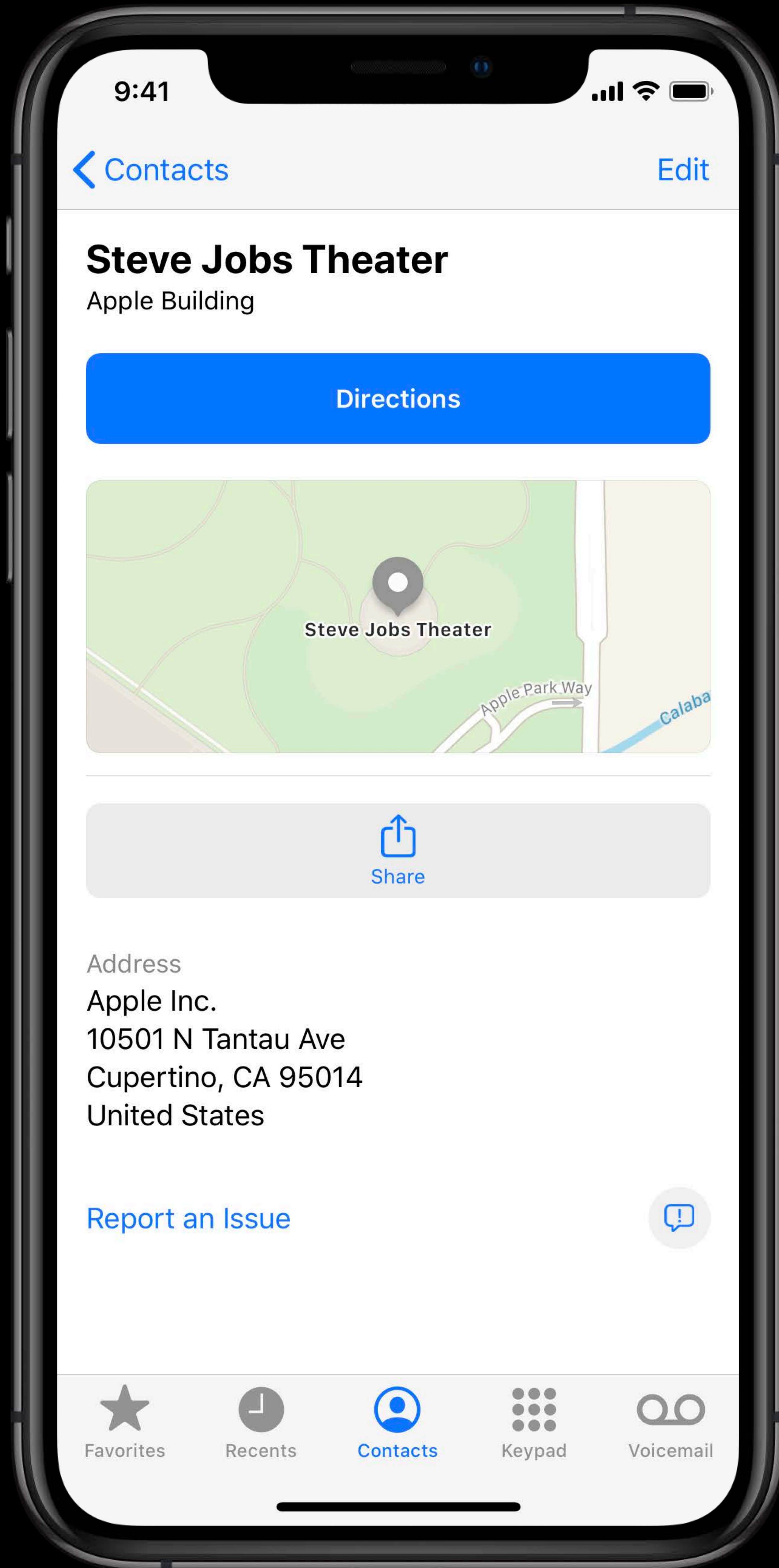
Contacts



Keypad

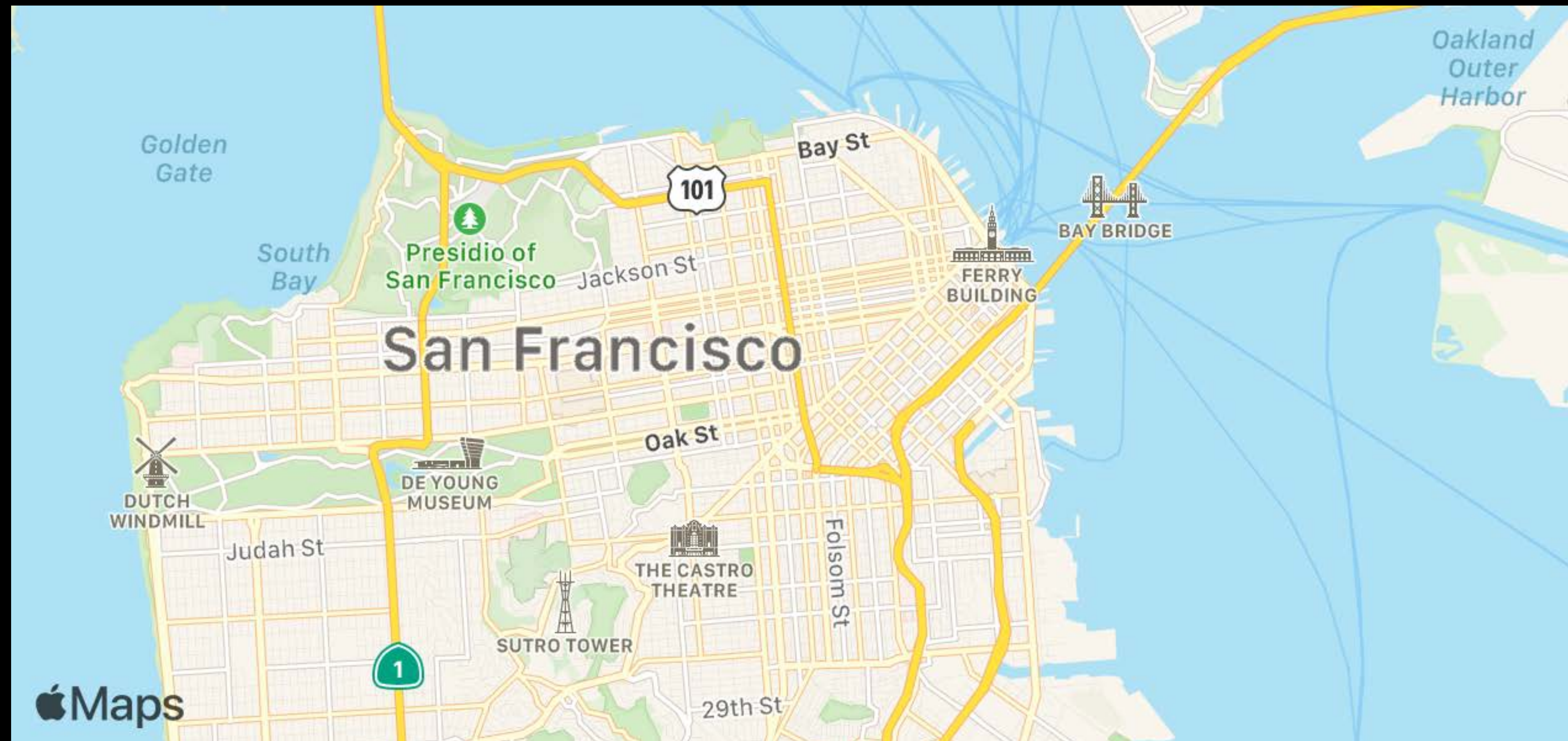


Voicemail



Maps Web Snapshots

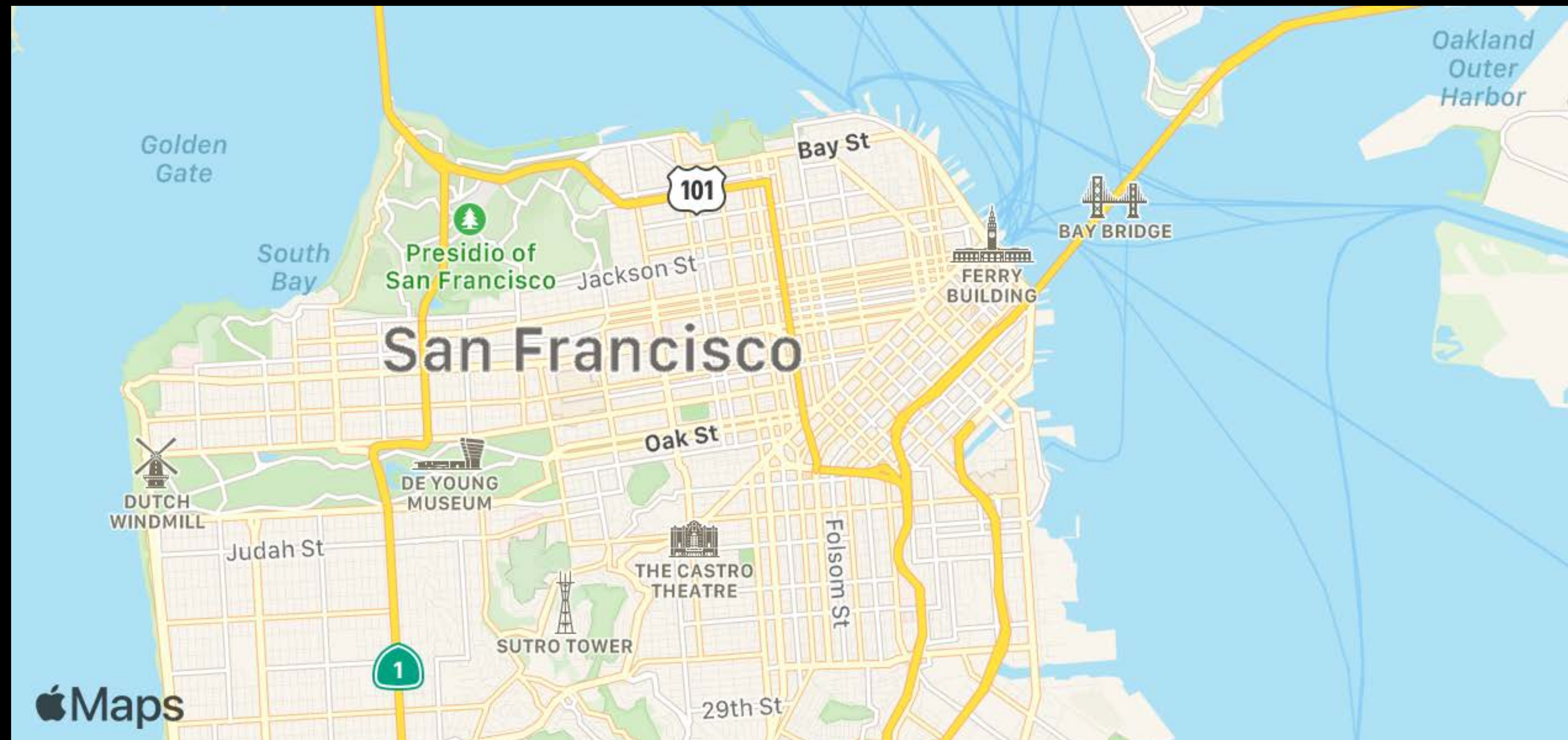
NEW



Maps Web Snapshots

NEW

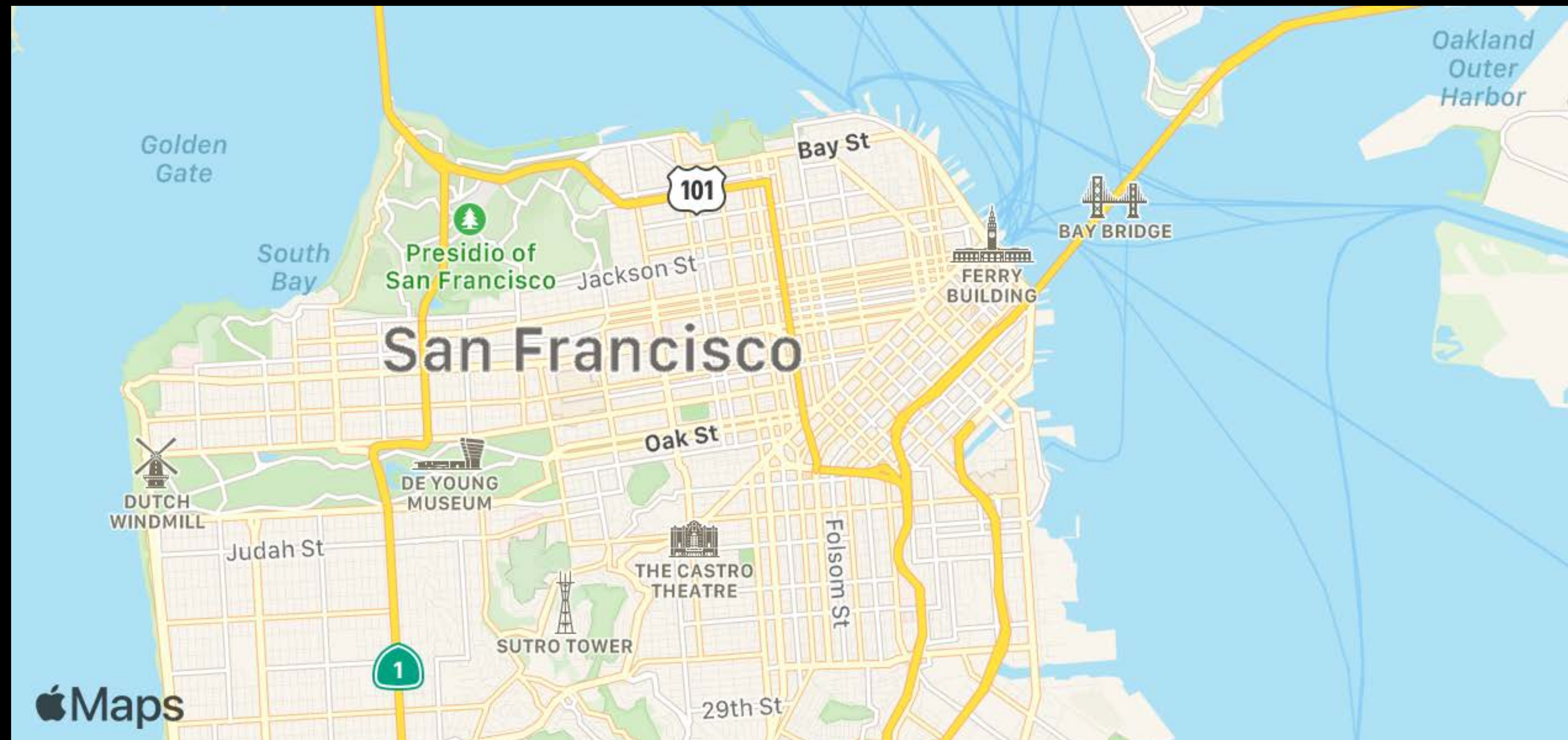
[snapshot.apple-mapkit.com/api/v1/snapshot?
center=37.78,-122.42&size=640x300](https://snapshot.apple-mapkit.com/api/v1/snapshot?center=37.78,-122.42&size=640x300)



Maps Web Snapshots

NEW

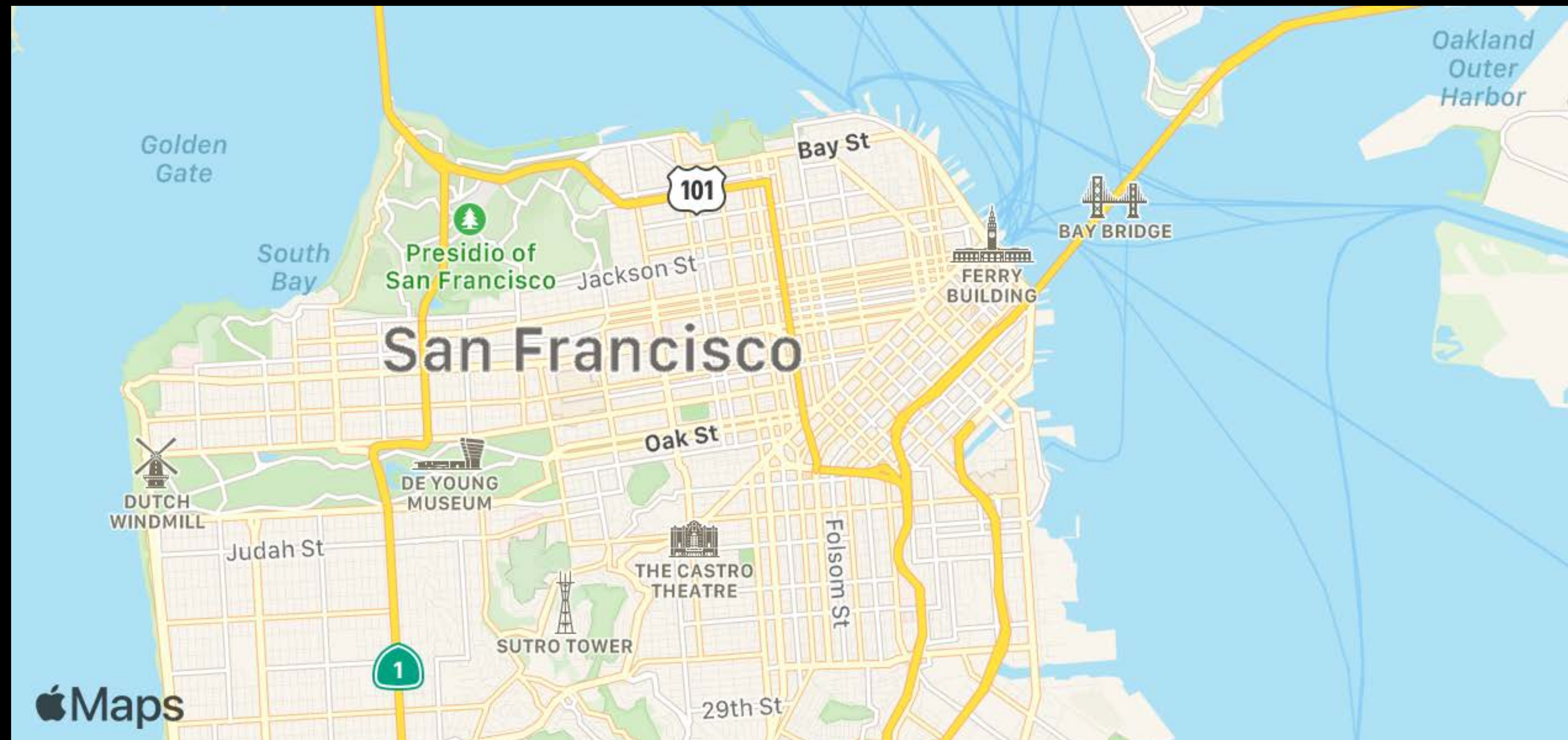
[snapshot.apple-mapkit.com/api/v1/snapshot?](https://snapshot.apple-mapkit.com/api/v1/snapshot?center=37.78,-122.42&size=640x300)
[center=37.78,-122.42&size=640x300](https://snapshot.apple-mapkit.com/api/v1/snapshot?center=37.78,-122.42&size=640x300)



Maps Web Snapshots

NEW

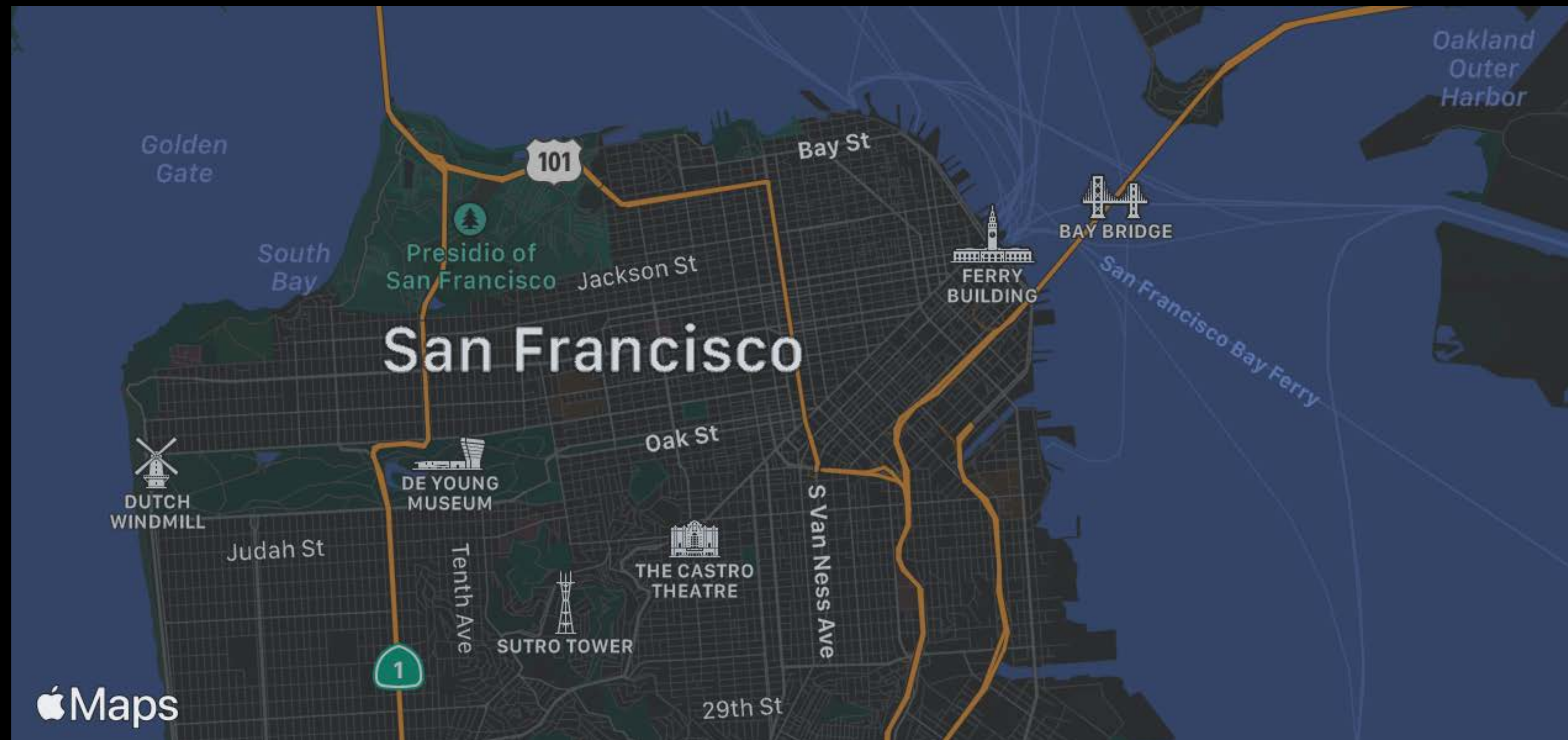
[snapshot.apple-mapkit.com/api/v1/snapshot?
center=37.78,-122.42&size=640x300](https://snapshot.apple-mapkit.com/api/v1/snapshot?center=37.78,-122.42&size=640x300)



Maps Web Snapshots

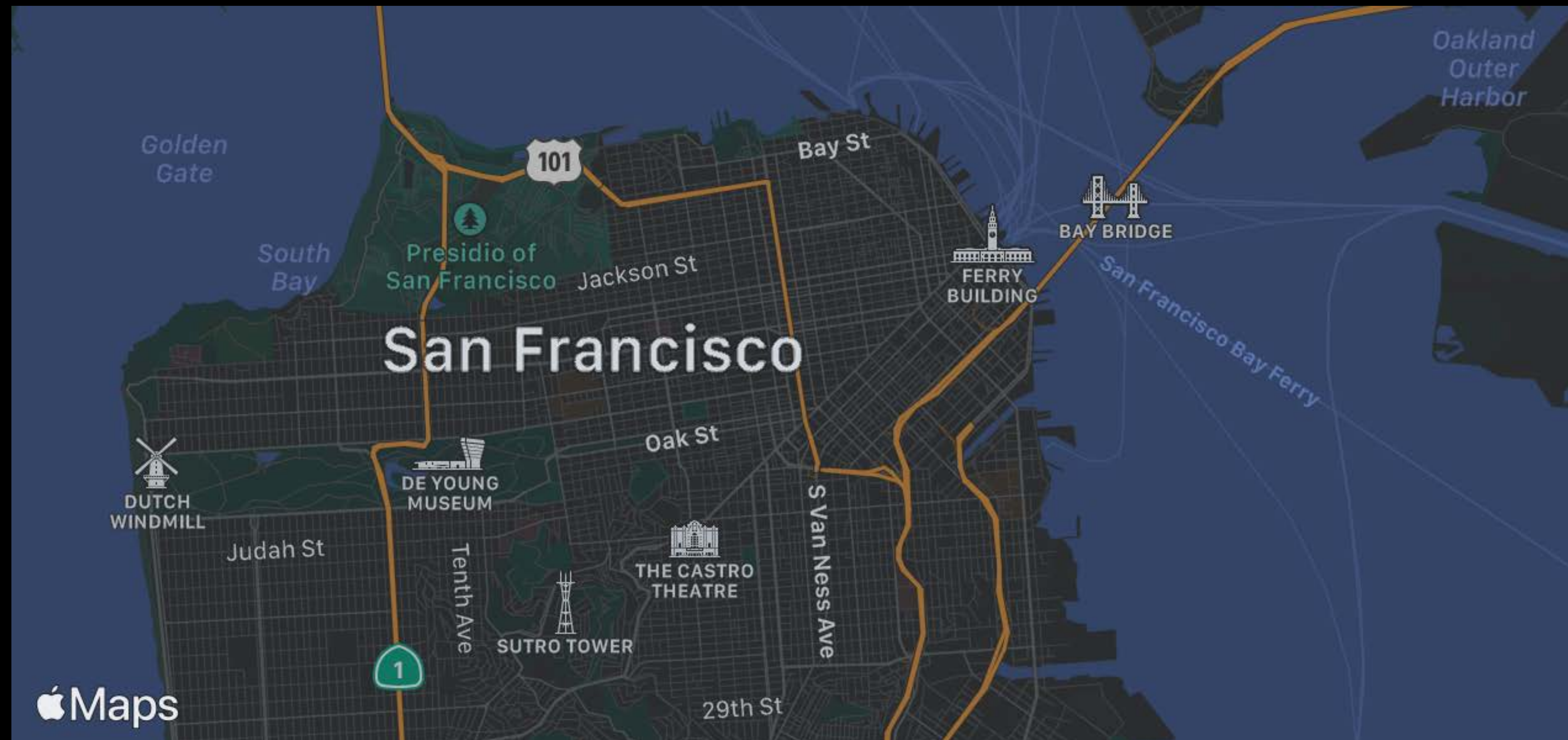
NEW

[snapshot.apple-mapkit.com/api/v1/snapshot?
center=37.78,-122.42&size=640x300&colorScheme=dark](https://snapshot.apple-mapkit.com/api/v1/snapshot?center=37.78,-122.42&size=640x300&colorScheme=dark)




Maps Web Snapshots

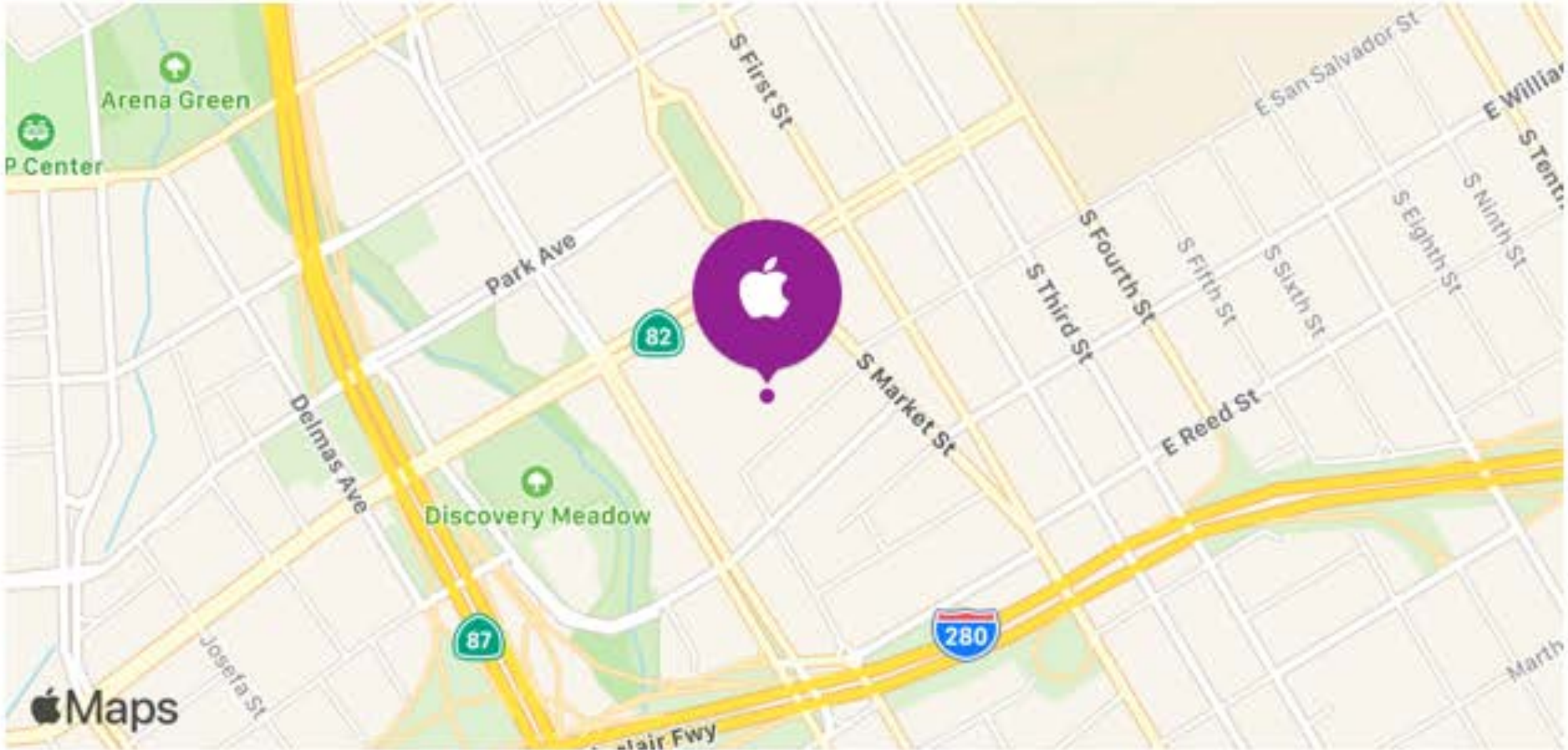
snapshot.apple-mapkit.com/api/v1/snapshot?
center=37.78,-122.42&size=640x300&colorScheme=dark
&signature=<yourSignatureHere>



Blair Lockhart
See You Soon
To: Teri Thomas



Inbox - Apple 4:49 PM 

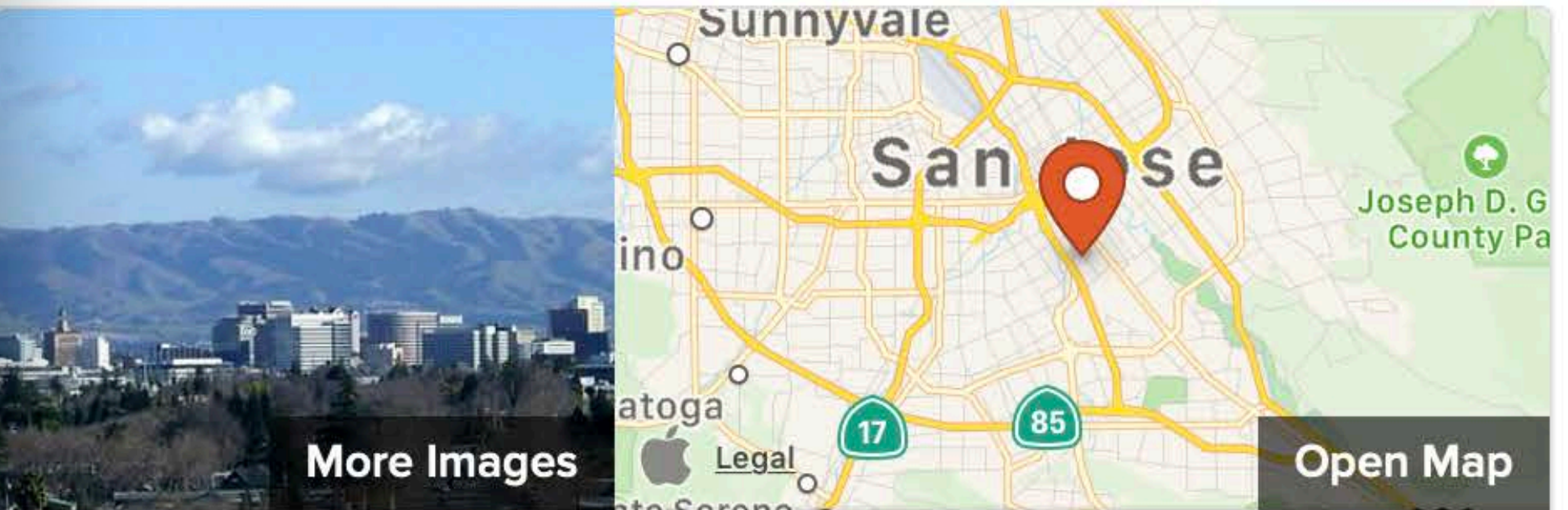
WWDC 2019
See You Soon.



Attending WWDC
Make sure your week goes smoothly with these essential details. Find out how to get to the conference, review important reminders, and more.

[Learn More >](#)




Privacy, simplified.  



San Jose, California
sanjoseca.gov

San Jose, officially the City of San José, is an economic, cultural, and political center of Silicon Valley and the largest city in Northern California. With an estimated 2017 population of 1,035,317, it is the third most populous city in California and the tenth most populous in United States. [Wikipedia](#)

CSA: San Jose-San Francisco-Oakland
Region: San Francisco Bay Area
Metro: San Jose-Sunnyvale-Santa Clara

   [Send Feedback](#)



MapKit JS

Create a Snapshot

Search for points of interest or directions to display on an Apple Maps Snapshot. Use the controls below to change characteristics of the snapshot such as dimension, language, scale, and color scheme, and click annotations or overlays to change their appearance.

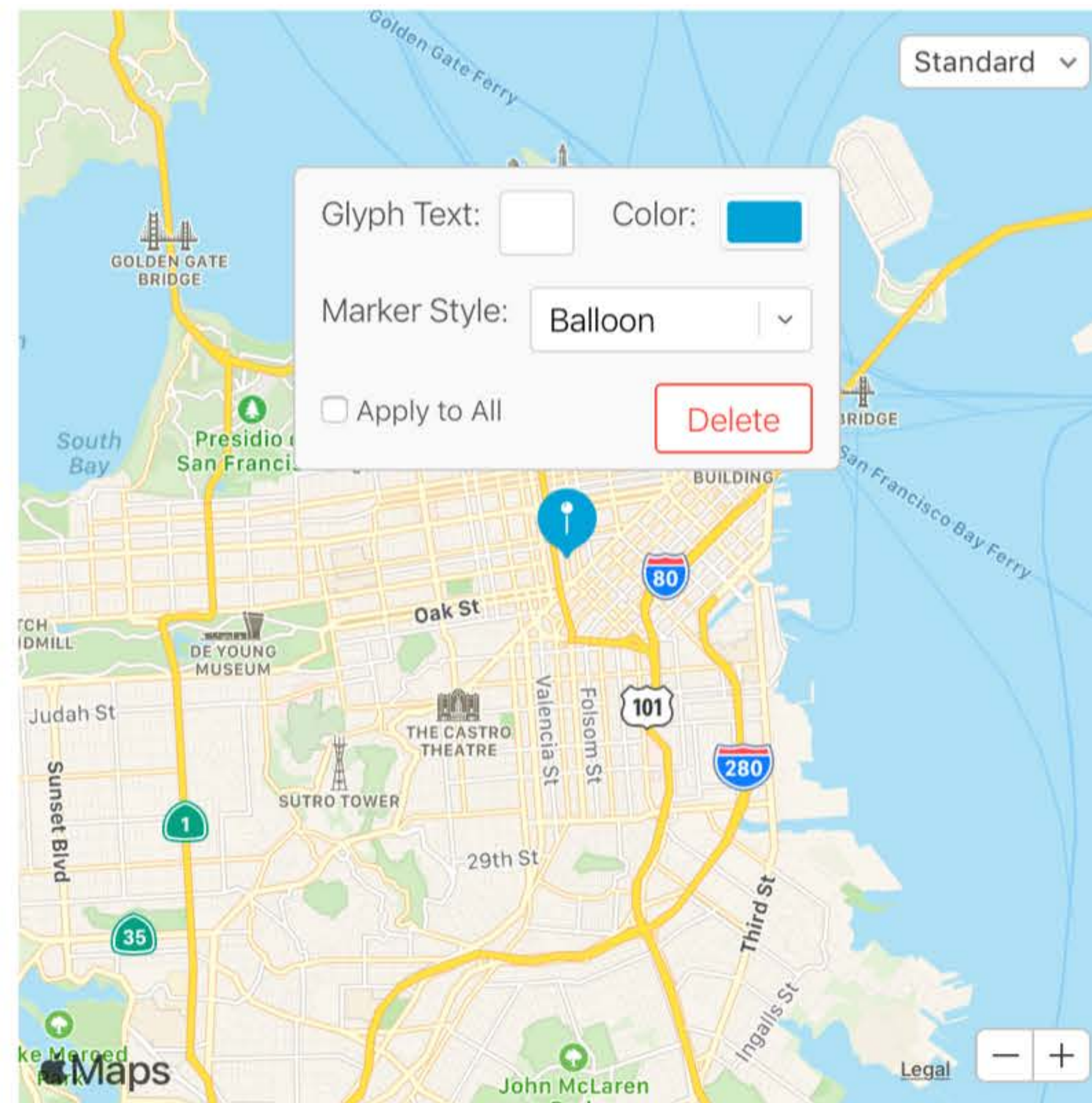
Search
San Francisco, CA, United States

Width: 500 Height: 500 Language: English (United States)

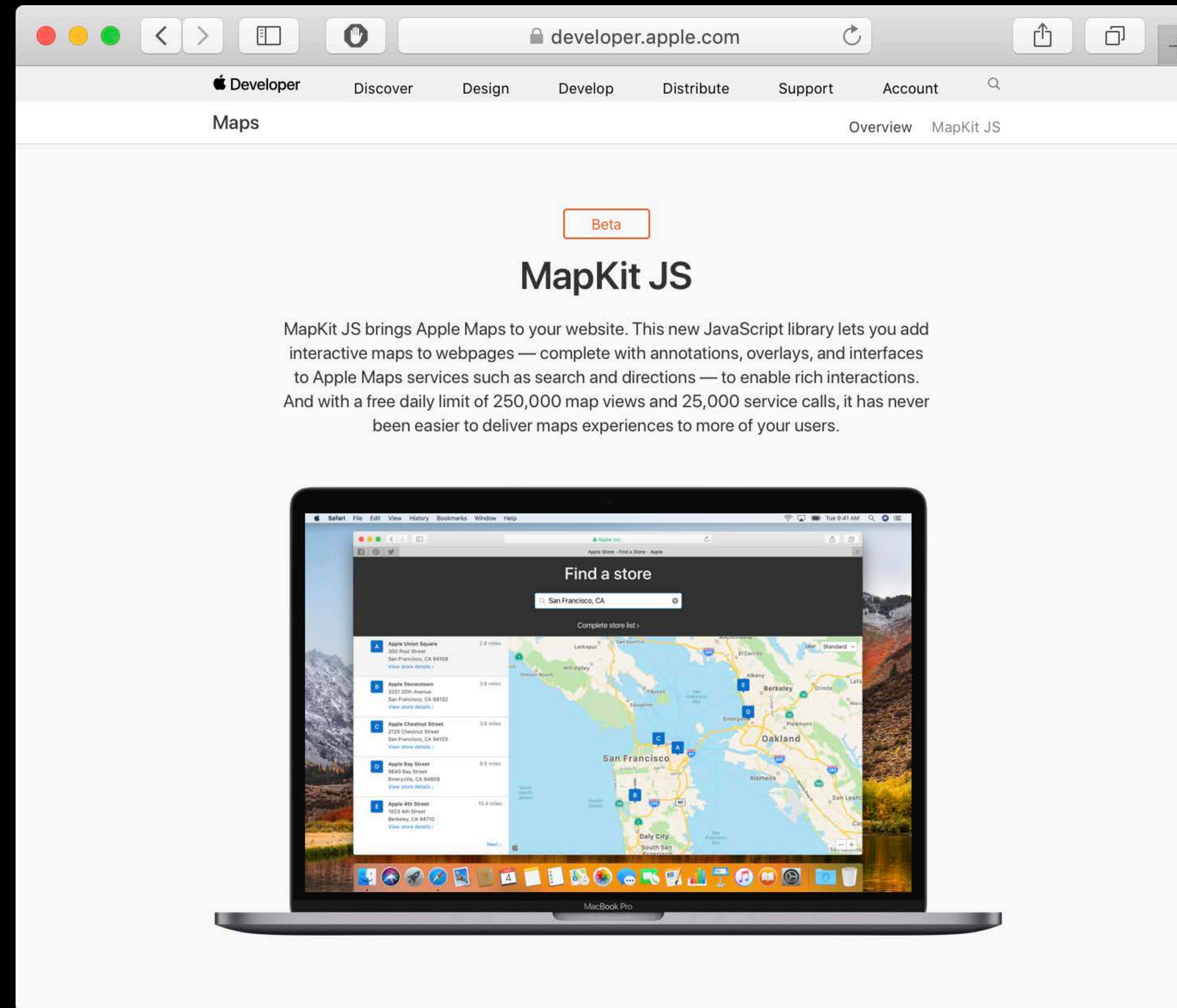
Dark Color Scheme Show Points of Interest Scale 2

[Show Snapshot](#) [Reset](#)

> Enter your credentials to create Snapshot URLs for your own use



developer.apple.com/mapkitjs



Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

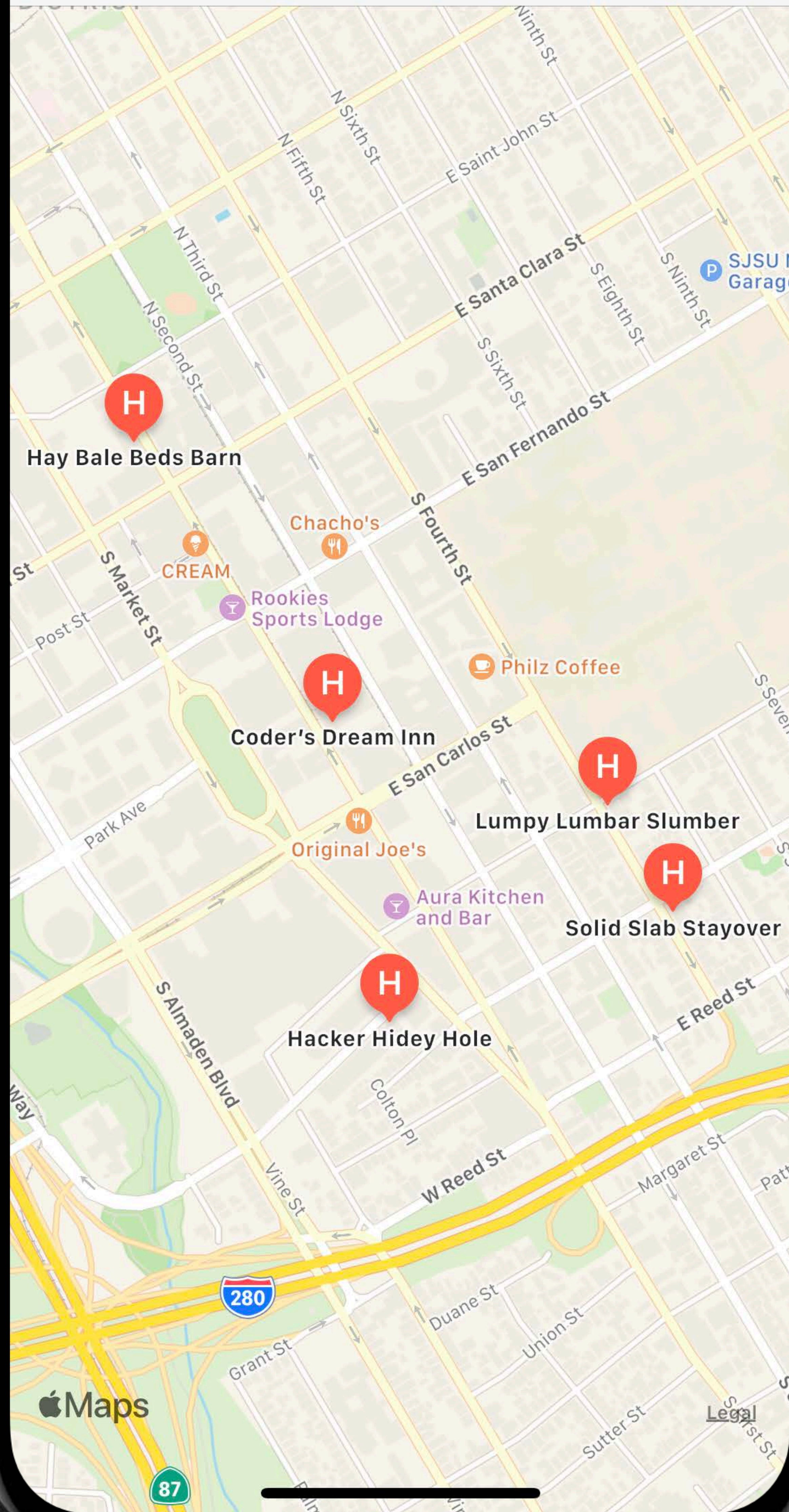
Taking control of the map view camera

9:41



[Back](#)

Accommodation



Apple Maps

87

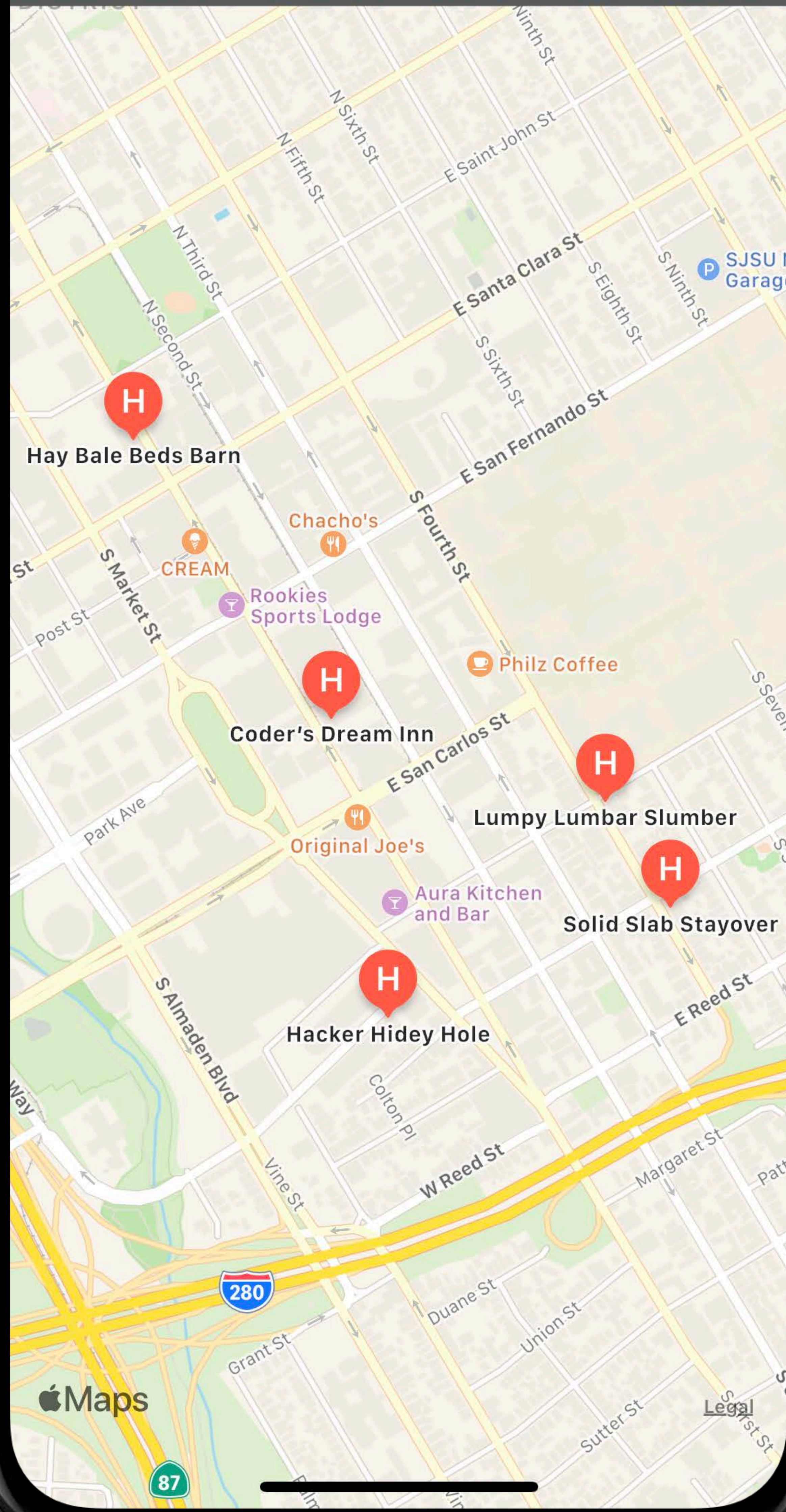
Legal

9:41



< Back

Accommodation



Apple Maps

87

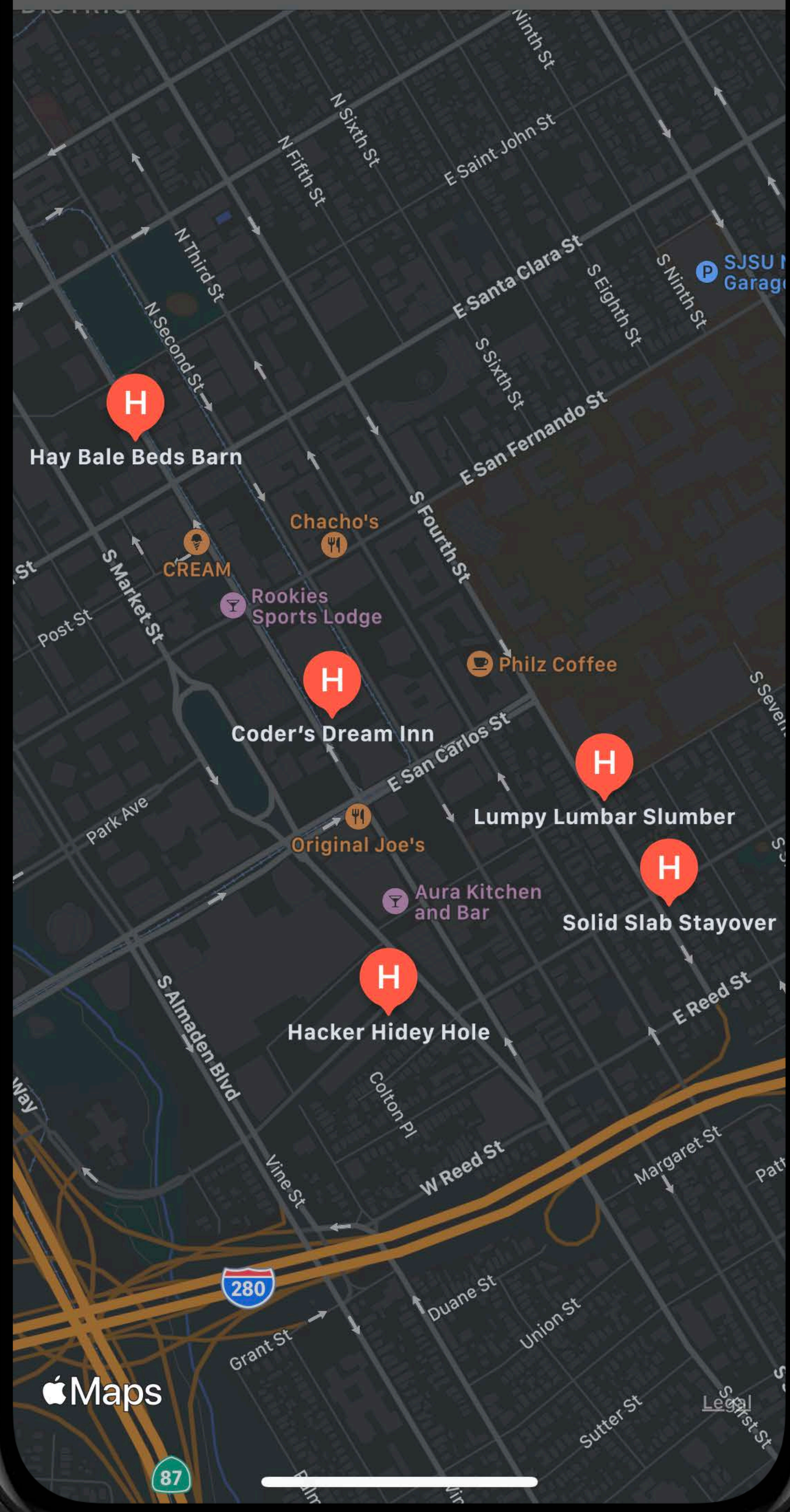
Legal

9:41



[Back](#)

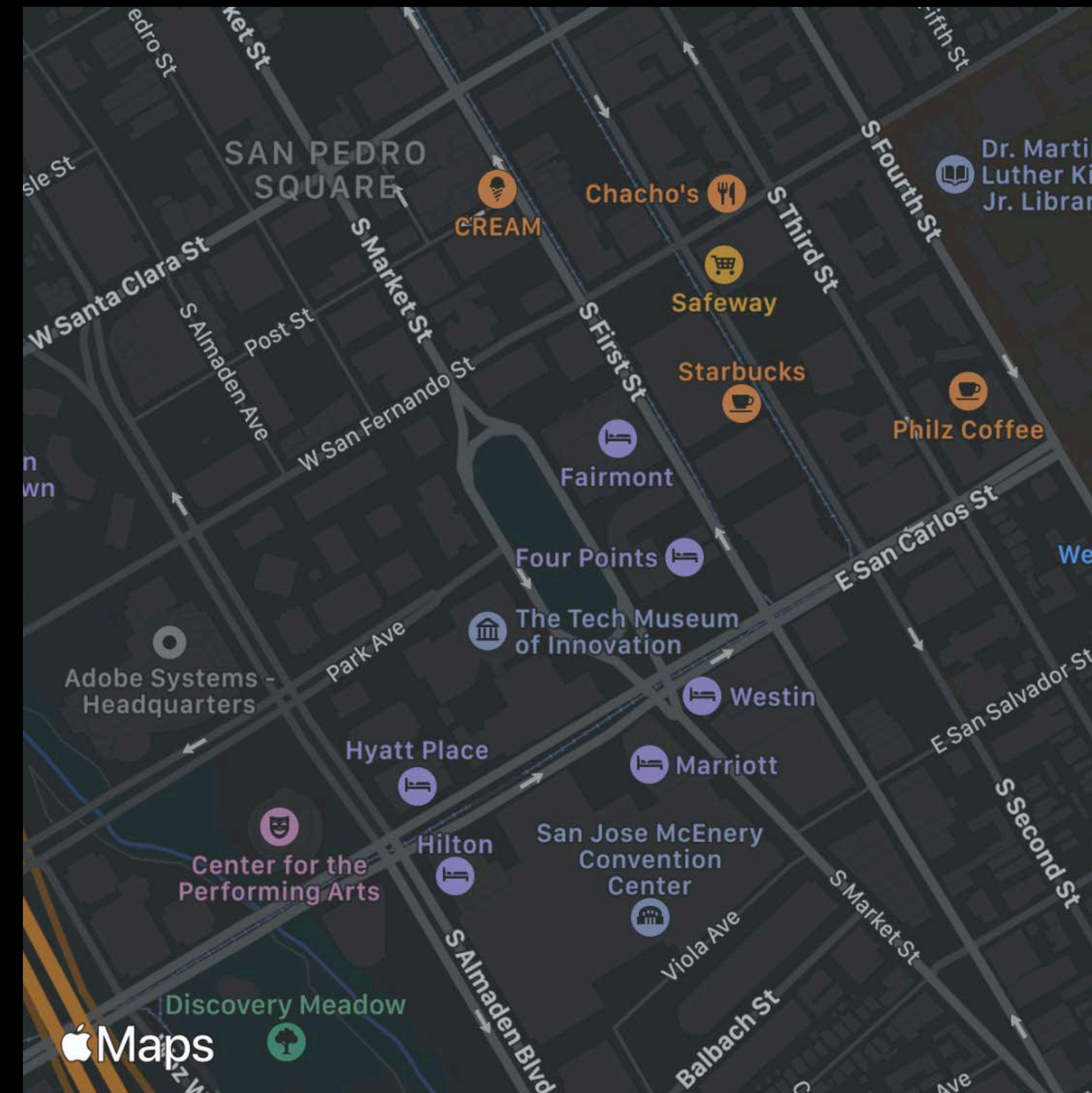
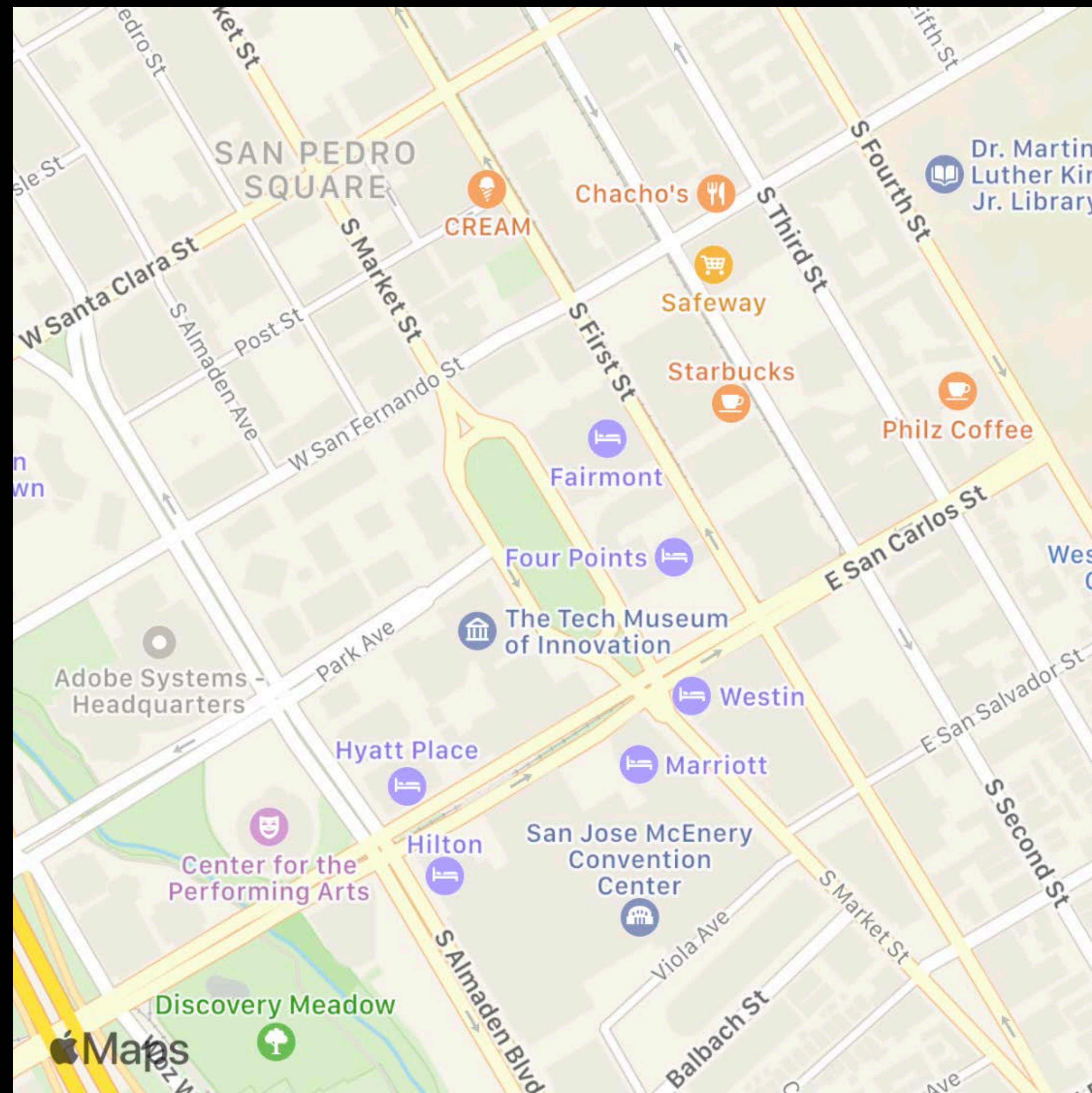
Accommodation



Apple Maps

MKMapSnapshotter

Maps without user interaction




```
// Create options
let options = MKMapSnapshotter.Options()
options.region = regionToSnapshot
options.size = CGSize(width: 300, height: 300)

// Alternative 1: Configure for view
options.traitCollection = snapshotView.traitCollection

// Alternative 2: Configure specific userInterfaceStyle
options.traitCollection = UITraitCollection(userInterfaceStyle: .dark)

// Create snapshot
let snapshotter = MKMapSnapshotter(options: options)
snapshotter.start { (snapshot, error) in
    // Handle snapshot if no error
}
}
```



```
// Create options
let options = MKMapSnapshotter.Options()
options.region = regionToSnapshot
options.size = CGSize(width: 300, height: 300)

// Alternative 1: Configure for view
options.traitCollection = snapshotView.traitCollection

// Alternative 2: Configure specific userInterfaceStyle
options.traitCollection = UITraitCollection(userInterfaceStyle: .dark)

// Create snapshot
let snapshotter = MKMapSnapshotter(options: options)
snapshotter.start { (snapshot, error) in
    // Handle snapshot if no error
}
}
```



```
// Create options
let options = MKMapSnapshotter.Options()
options.region = regionToSnapshot
options.size = CGSize(width: 300, height: 300)

// Alternative 1: Configure for view
options.traitCollection = snapshotView.traitCollection

// Alternative 2: Configure specific userInterfaceStyle
options.traitCollection = UITraitCollection(userInterfaceStyle: .dark)

// Create snapshot
let snapshotter = MKMapSnapshotter(options: options)
snapshotter.start { (snapshot, error) in
    // Handle snapshot if no error
}
}
```



```
// Create options
let options = MKMapSnapshotter.Options()
options.region = regionToSnapshot
options.size = CGSize(width: 300, height: 300)

// Alternative 1: Configure for view
options.traitCollection = snapshotView.traitCollection

// Alternative 2: Configure specific userInterfaceStyle
options.traitCollection = UITraitCollection(userInterfaceStyle: .dark)

// Create snapshot
let snapshotter = MKMapSnapshotter(options: options)
snapshotter.start { (snapshot, error) in
    // Handle snapshot if no error
}
}
```



```
// Create options
let options = MKMapSnapshotter.Options()
options.region = regionToSnapshot
options.size = CGSize(width: 300, height: 300)

// Alternative 1: Configure for view
options.traitCollection = snapshotView.traitCollection

// Alternative 2: Configure specific userInterfaceStyle
options.traitCollection = UITraitCollection(userInterfaceStyle: .dark)

// Create snapshot
let snapshotter = MKMapSnapshotter(options: options)
snapshotter.start { (snapshot, error) in
    // Handle snapshot if no error
}
}
```



```
// Create options
let options = MKMapSnapshotter.Options()
options.region = regionToSnapshot
options.size = CGSize(width: 300, height: 300)

// Alternative 1: Configure for view
options.traitCollection = snapshotView.traitCollection

// Alternative 2: Configure specific userInterfaceStyle
options.traitCollection = UITraitCollection(userInterfaceStyle: .dark)

// Create snapshot
let snapshotter = MKMapSnapshotter(options: options)
snapshotter.start { (snapshot, error) in
    // Handle snapshot if no error
}
```



```
// Create options
let options = MKMapSnapshotter.Options()
options.region = regionToSnapshot
options.size = CGSize(width: 300, height: 300)

// Alternative 1: Configure for view
options.traitCollection = snapshotView.traitCollection

// Alternative 2: Configure specific userInterfaceStyle
options.traitCollection = UITraitCollection(userInterfaceStyle: .dark)

// Create snapshot
let snapshotter = MKMapSnapshotter(options: options)
snapshotter.start { (snapshot, error) in
    // Handle snapshot if no error
}
}
```



```
// Create options
let options = MKMapSnapshotter.Options()
options.region = regionToSnapshot
options.size = CGSize(width: 300, height: 300)

// Alternative 1: Configure for view
options.traitCollection = snapshotView.traitCollection

// Alternative 2: Configure specific userInterfaceStyle
options.traitCollection = UITraitCollection(userInterfaceStyle: .dark)

// Create snapshot
let snapshotter = MKMapSnapshotter(options: options)
snapshotter.start { (snapshot, error) in
    // Handle snapshot if no error
}
}
```


Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

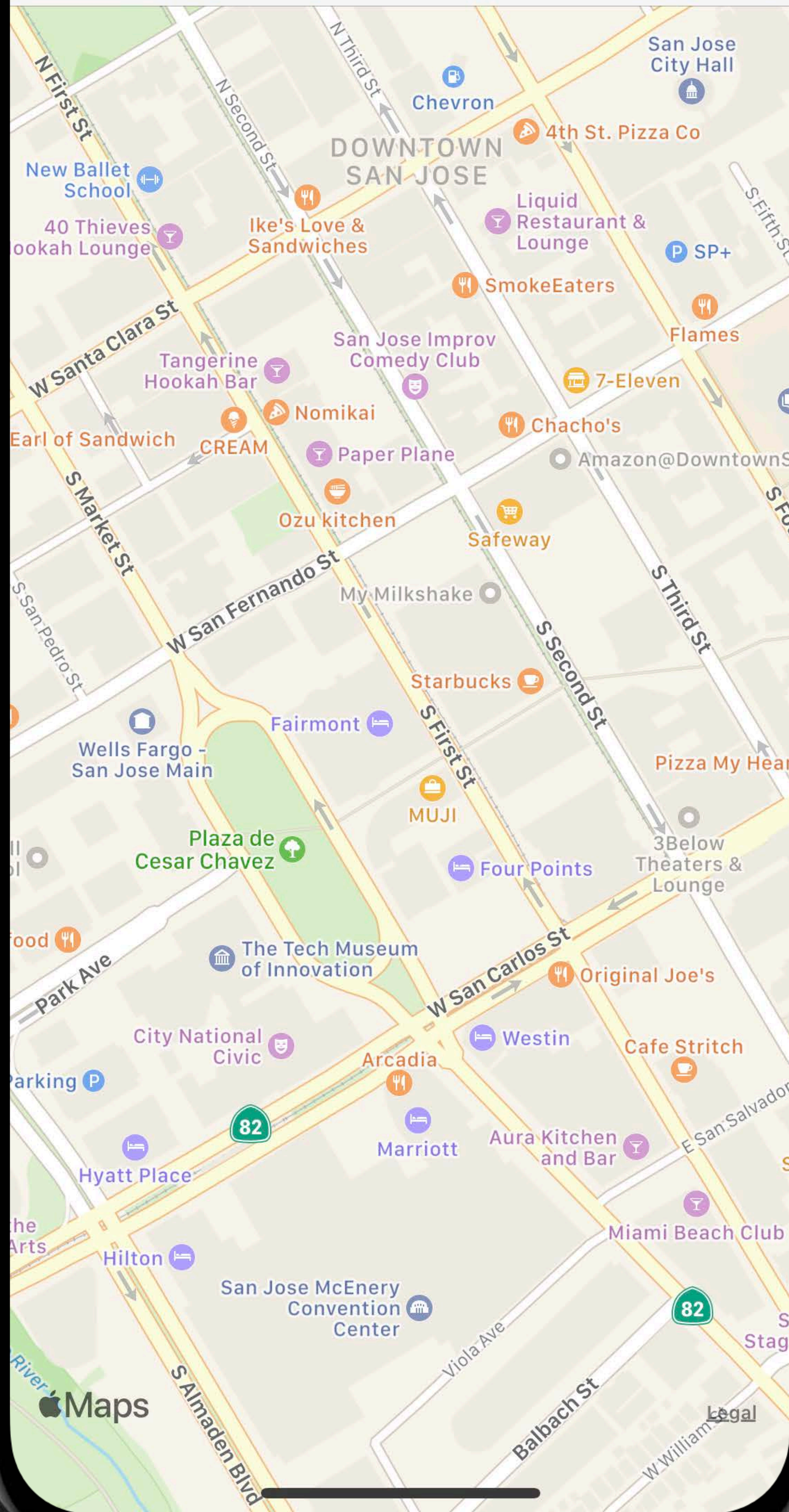
Taking control of the map view camera

9:41



[Back](#)

Accommodation



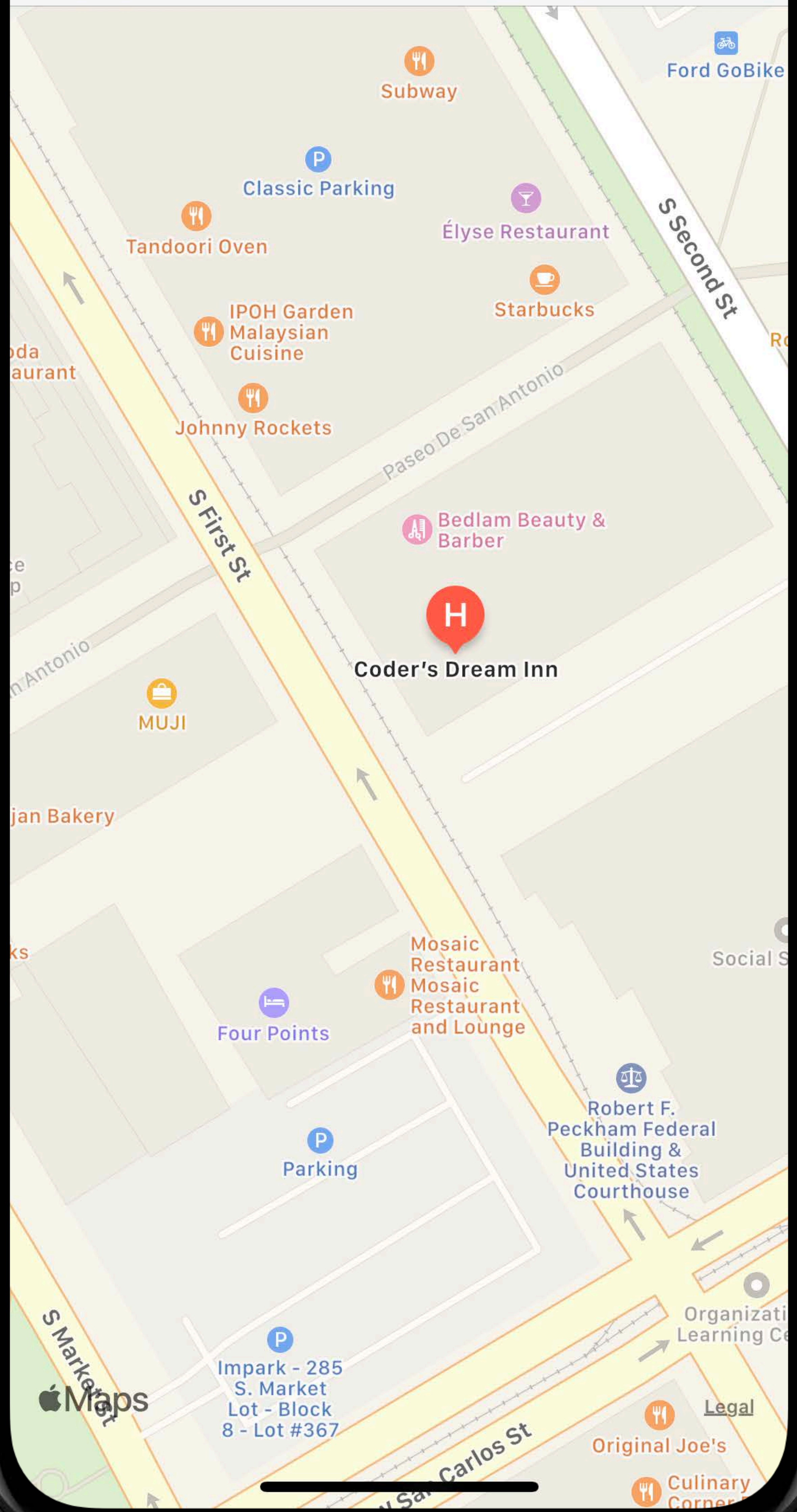
Apple Maps

9:41



[Back](#)

Accommodation



Apple Maps

Impark - 285
S. Market
Lot - Block
8 - Lot #367

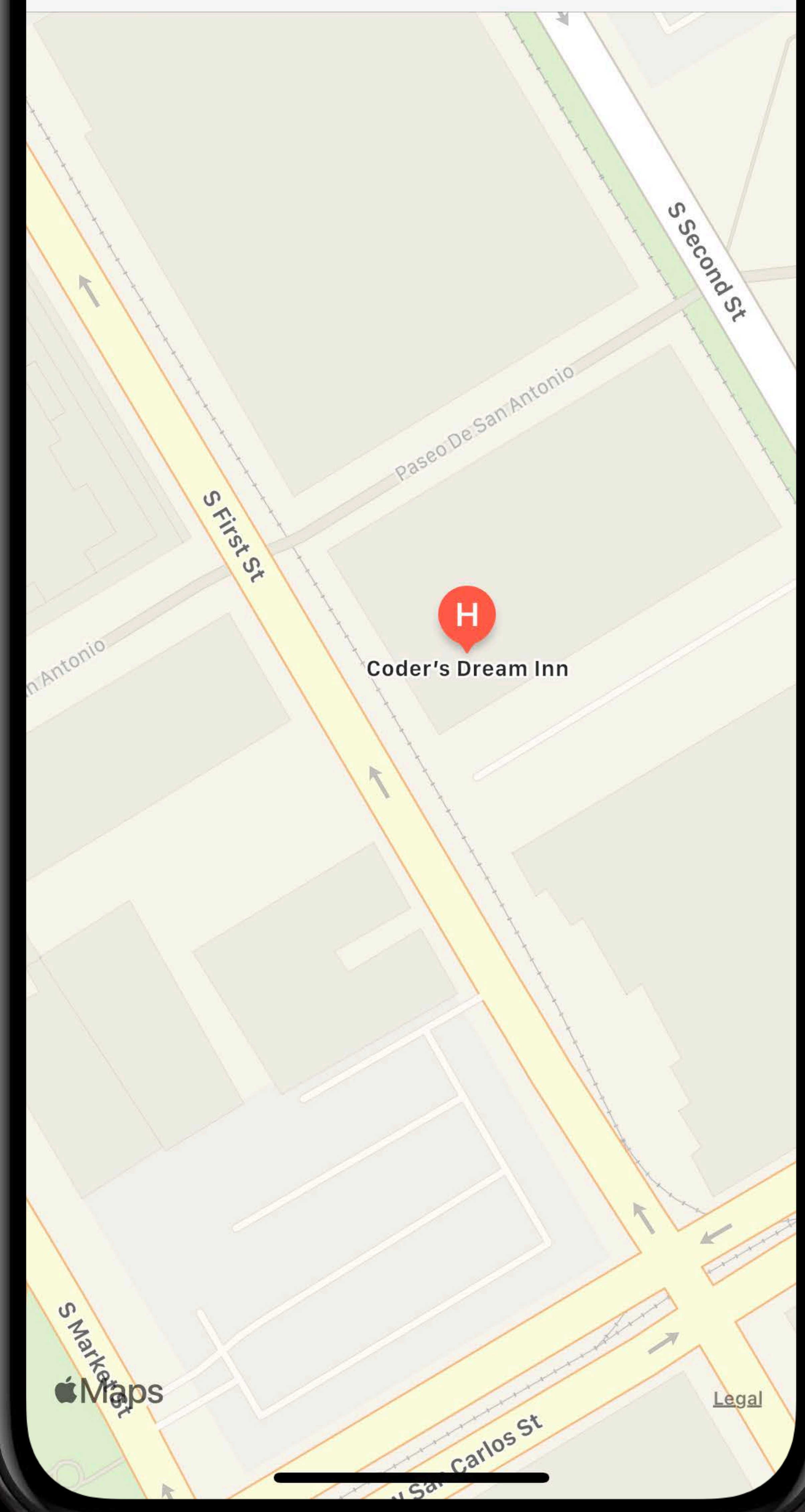
Carlos St

9:41



[Back](#)

Accommodation



Apple Maps

Legal

MKPointOfInterestCategory

NEW

.airport

.amusementPark

.aquarium

.atm

.bakery

.bank

.beach

.brewery

.cafe

.campground

.carRental

.evCharger

.fireStation

.fitnessCenter

.foodMarket

.gasStation

.hospital

.hotel

.laundry

.library

.marina

.movieTheater

.museum

.nationalPark

.nightlife

.park

.parking

.pharmacy

.playground

.police

.postOffice

.publicTransport

.religiousSit

.restaurant

.restroom

.school

.stadium

.store

.theater

.university

.winery

.zoo

MKPointOfInterestFilter

NEW

MKPointOfInterestFilter



NEW

Include specific categories

Exclude specific categories

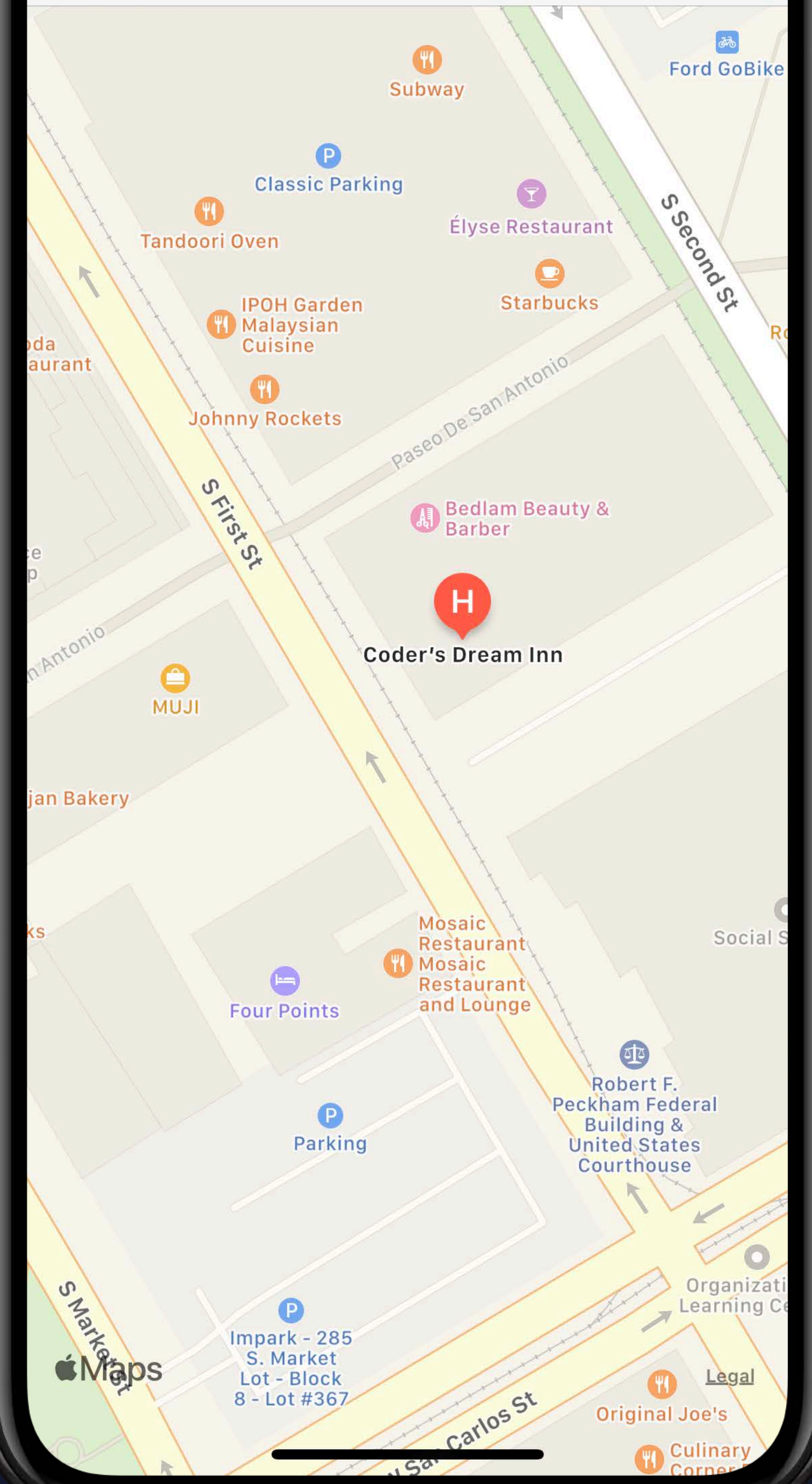
Exclude all categories

9:41



[Back](#)

Accommodation



Apple Maps

Impark - 285
S. Market
Lot - Block
8 - Lot #367

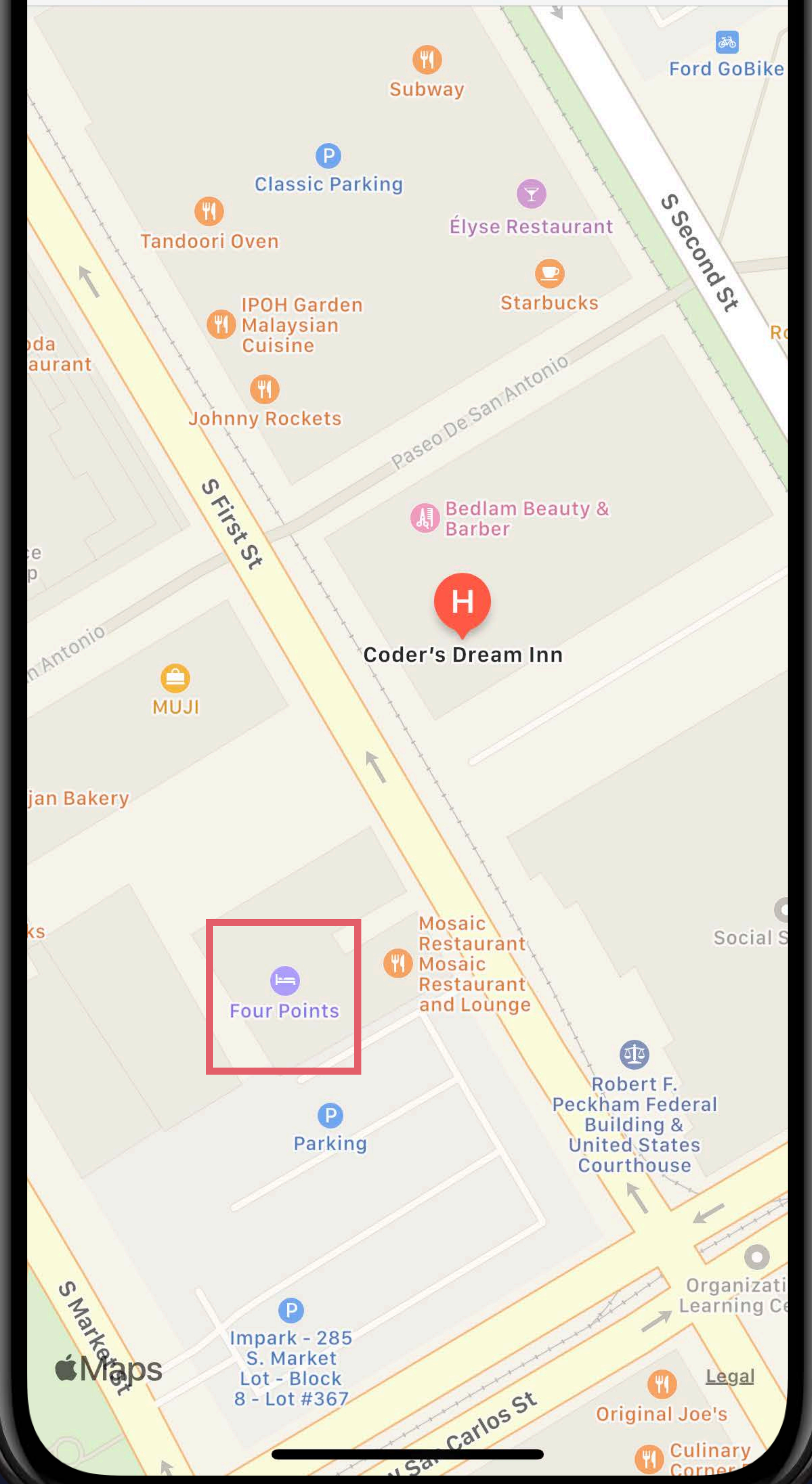
Carlos St

9:41



[Back](#)

Accommodation

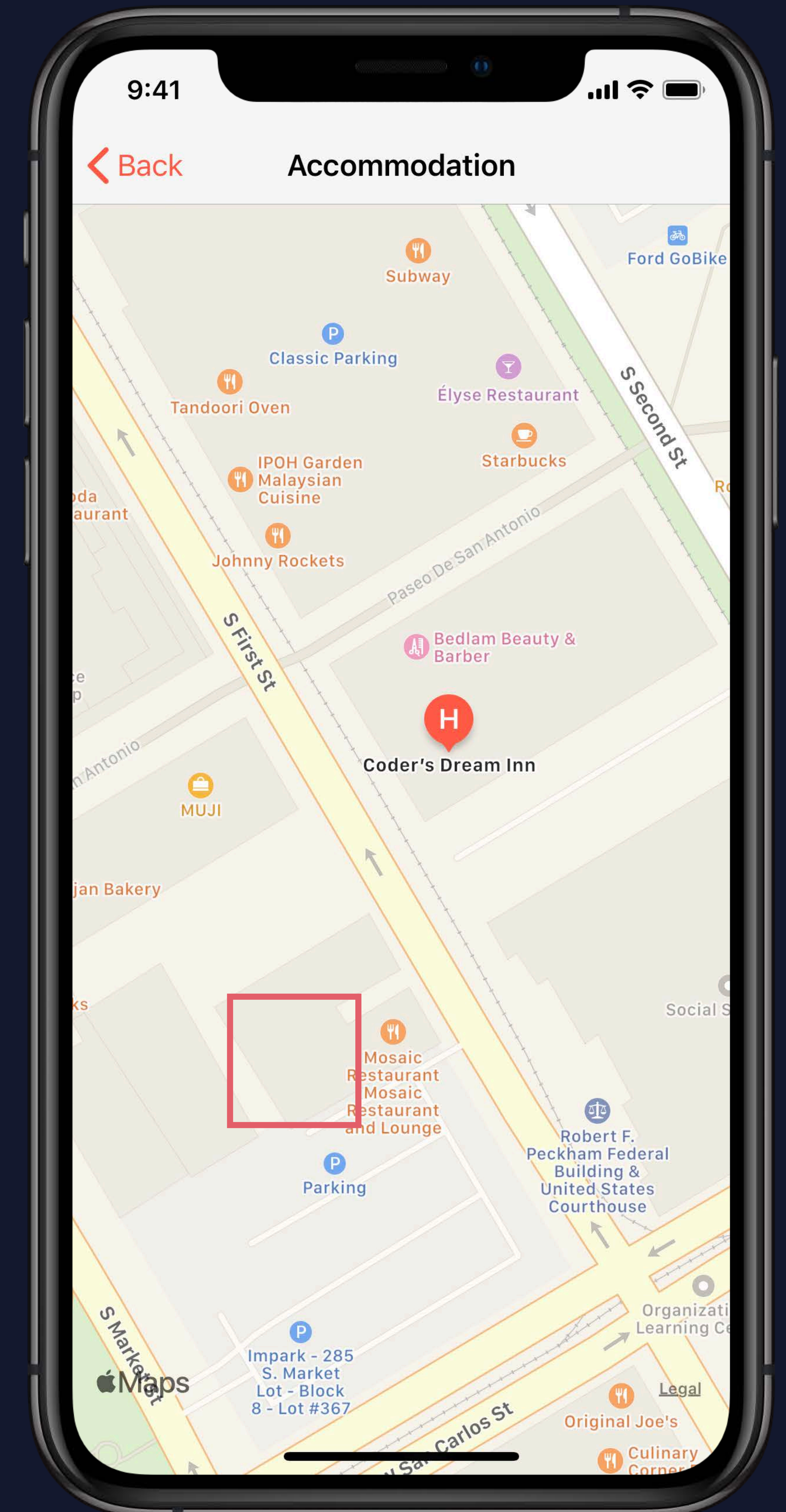



```
// Exclusion Filtering
```

```
let categories = [ .hotel ]
```

```
let filter = MKPointOfInterestFilter(excluding: categories)
```

```
mapView.pointOfInterestFilter = filter
```

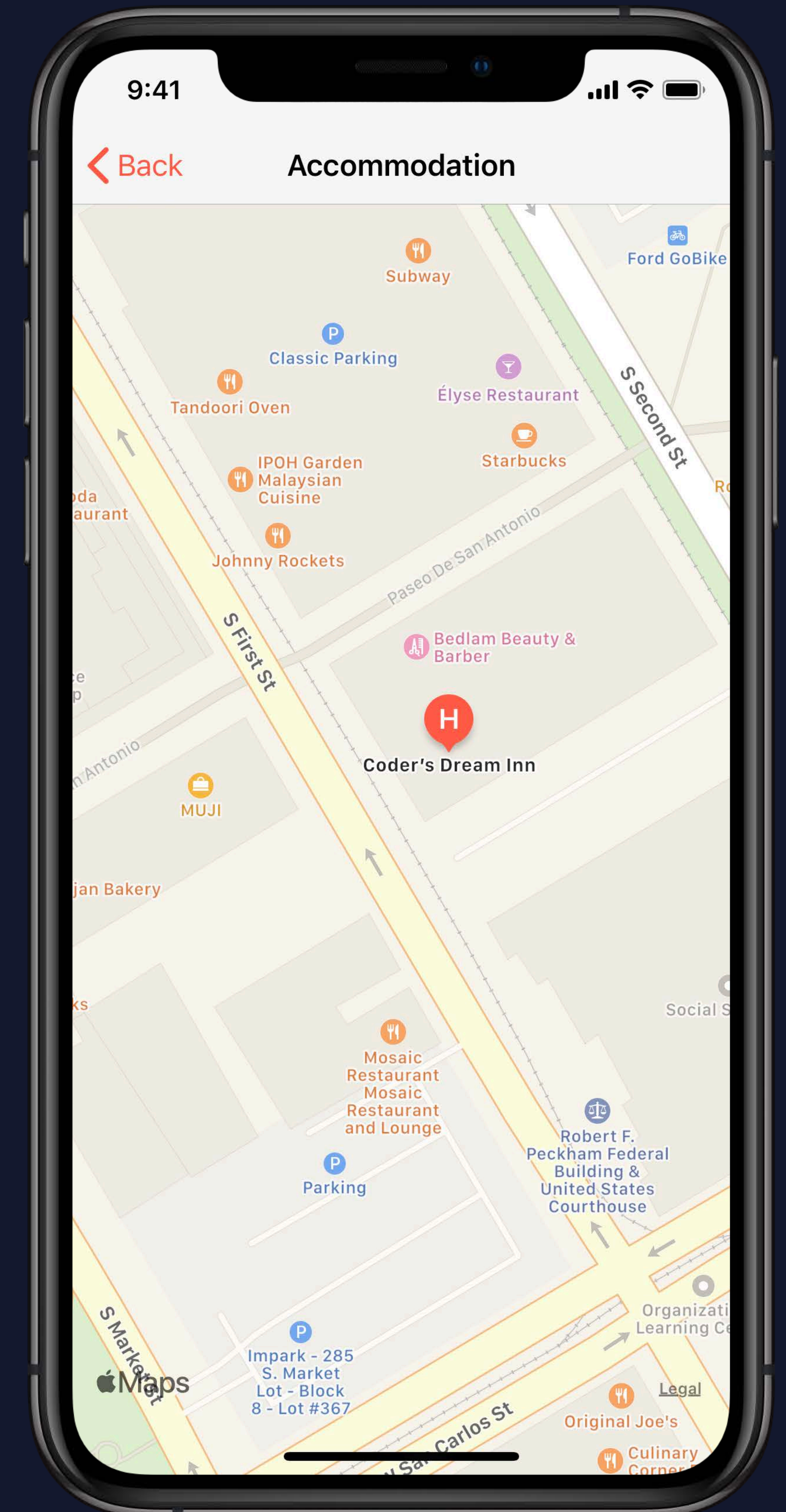



```
// Inclusion Filtering
```

```
let categories = [ .restaurant, .cafe, .nightlife, .parking ]
```

```
let filter = MKPointOfInterestFilter(including: categories)
```

```
mapView.pointOfInterestFilter = filter
```

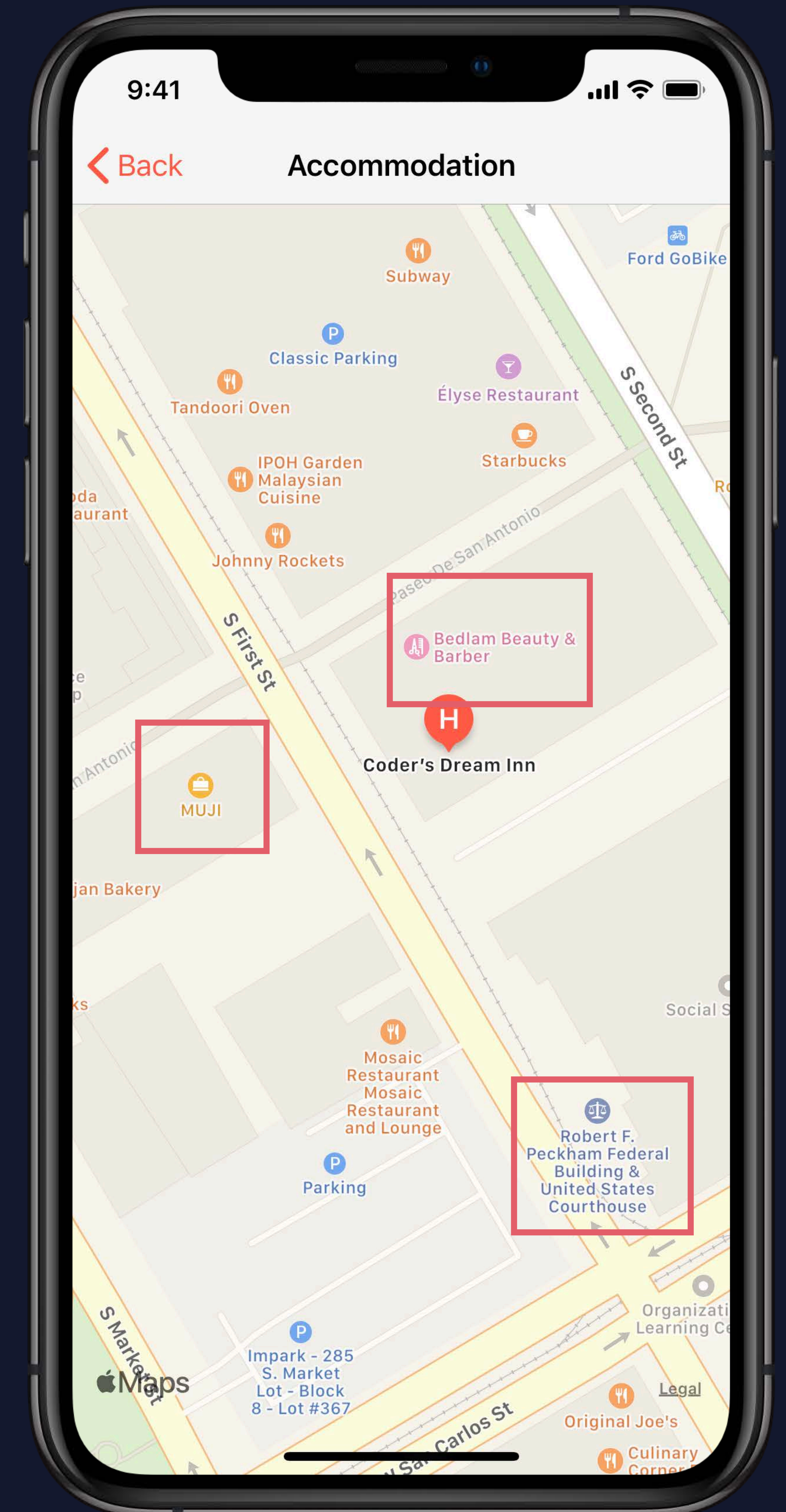



```
// Inclusion Filtering
```

```
let categories = [ .restaurant, .cafe, .nightlife, .parking ]
```

```
let filter = MKPointOfInterestFilter(including: categories)
```

```
mapView.pointOfInterestFilter = filter
```

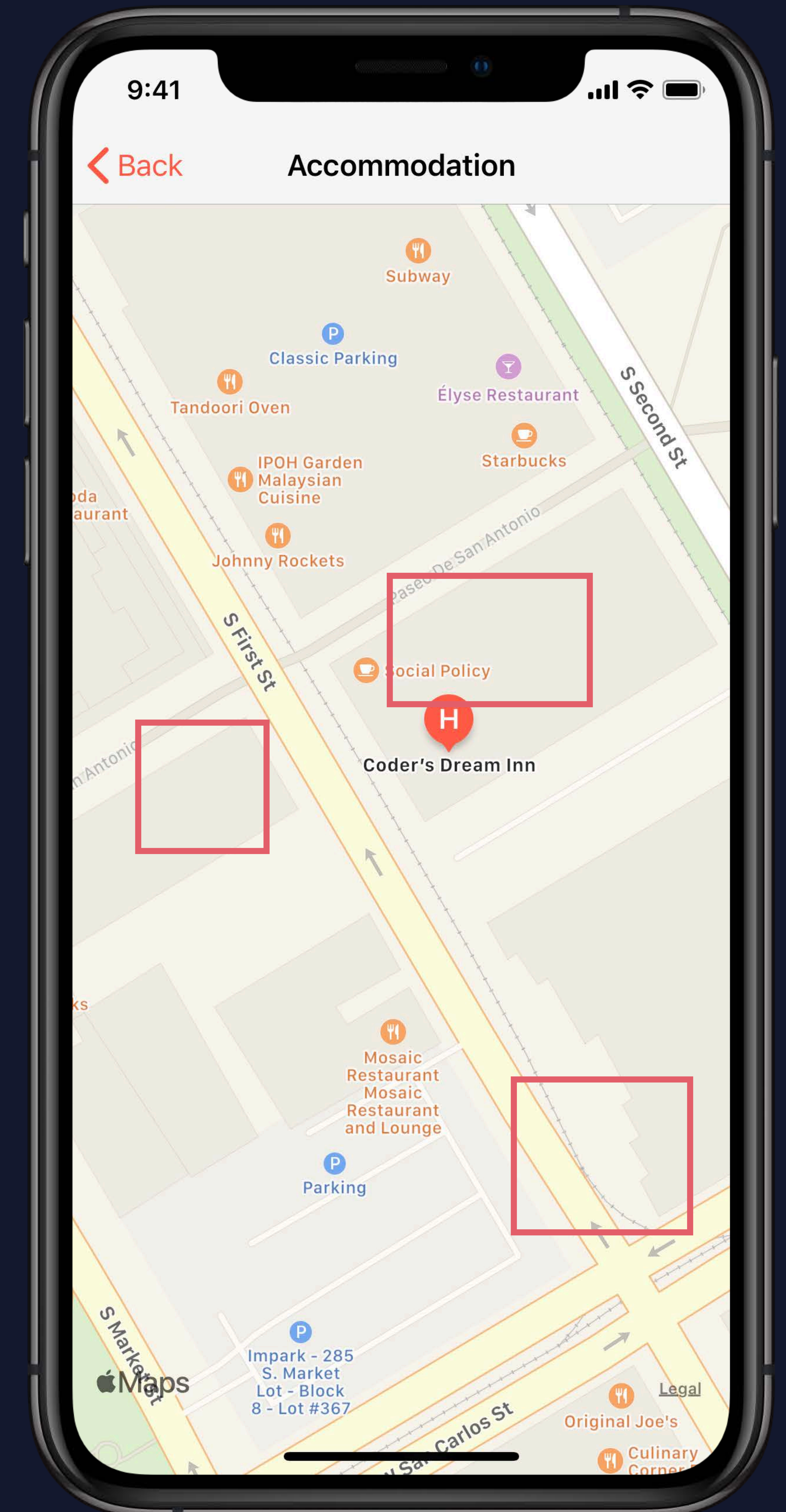



```
// Inclusion Filtering
```

```
let categories = [ .restaurant, .cafe, .nightlife, .parking ]
```

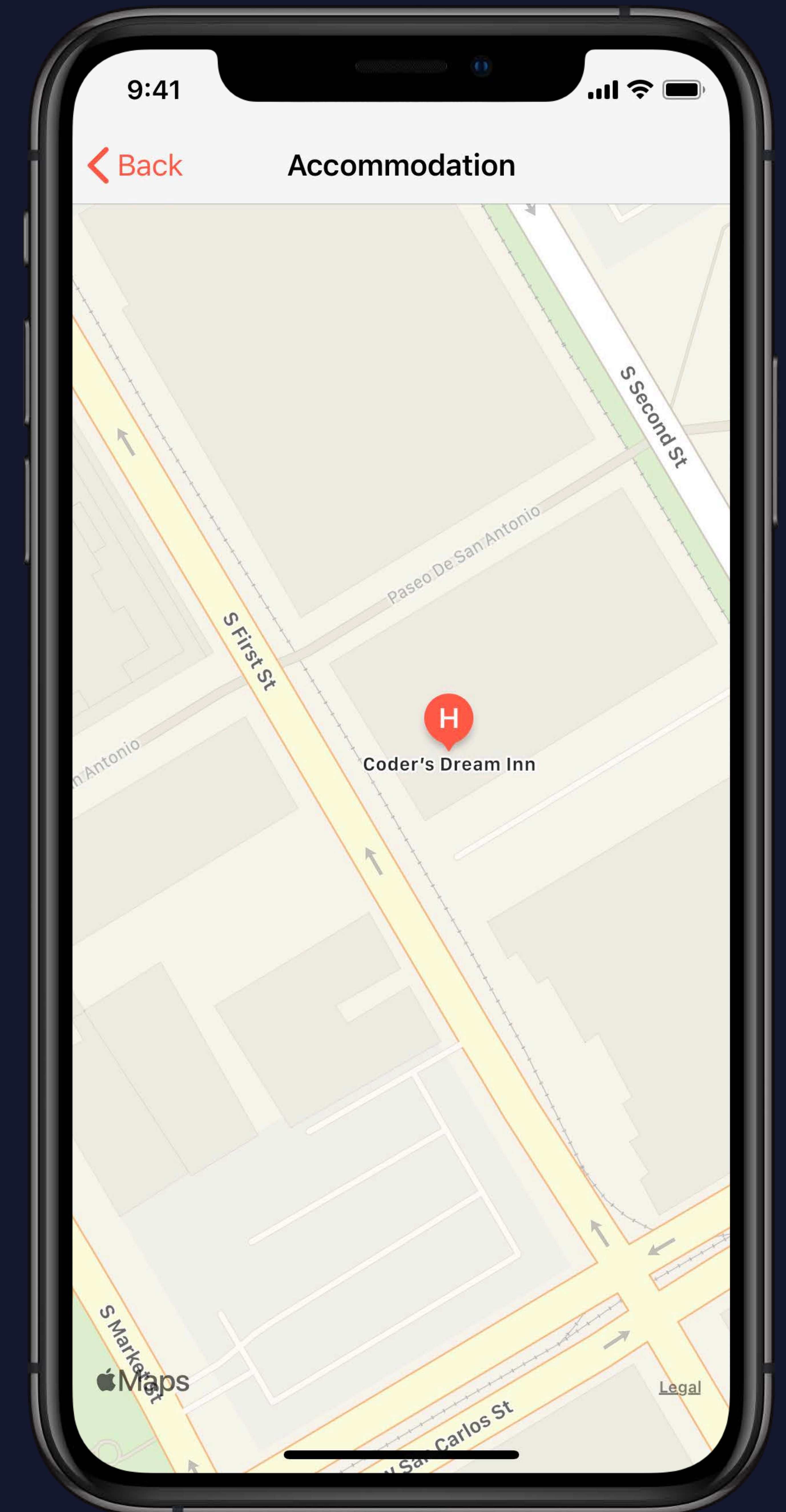
```
let filter = MKPointOfInterestFilter(including: categories)
```

```
mapView.pointOfInterestFilter = filter
```




```
// Excluding All Points of Interest
```

```
let filter = MKPointOfInterestFilter.excludingAll  
mapView.pointOfInterestFilter = filter
```



Point of Interest Filtering in MapKit JS



NEW

Coming this fall!

```
// Exclusion Filtering
let filter = mapkit.PointOfInterestFilter.excluding([ mapkit.PointOfInterestCategory.Hotel ]);

// Inclusion Filtering
let filter = mapkit.PointOfInterestFilter.including([ mapkit.PointOfInterestCategory.Hotel ]);

map.pointOfInterestFilter = filter;
```


Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

9:41

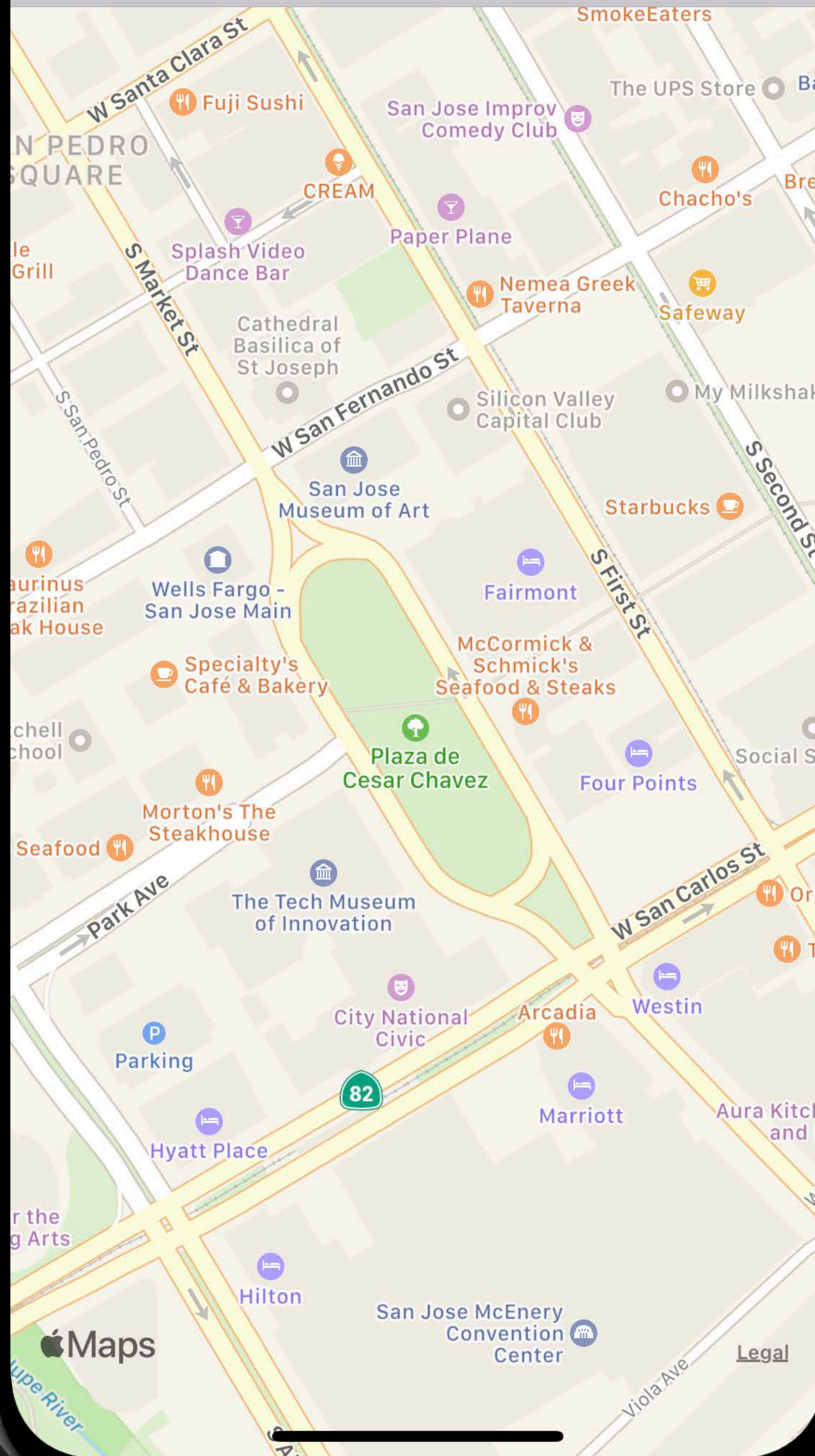


< Back

After Hours

Search for restaurants & bars

Cancel



Apple Maps

Legal

9:41



< Back

After Hours

Q Mor|



Cancel

Morgan Hill, CA, United States

Morton's The Steakhouse

177 Park Ave, San Jose, CA 95113, United States

Morris Dailey Auditorium - San Jose State Uni...

1 Washington Sq, San Jose, CA 95192, United States

Morphe

2855 Stevens Creek Blvd, Santa Clara, CA 95050-6709, Unite...

Morrill Middle School

1970 Morrill Ave, San Jose, CA 95132, United States

Morrison Park Dr

San Jose, CA, United States

Morse St

San Jose, CA, United States

q w e r t y u i o p

a s d f g h j k l

⬆ z x c v b n m ⬇

123

space

Search



9:41



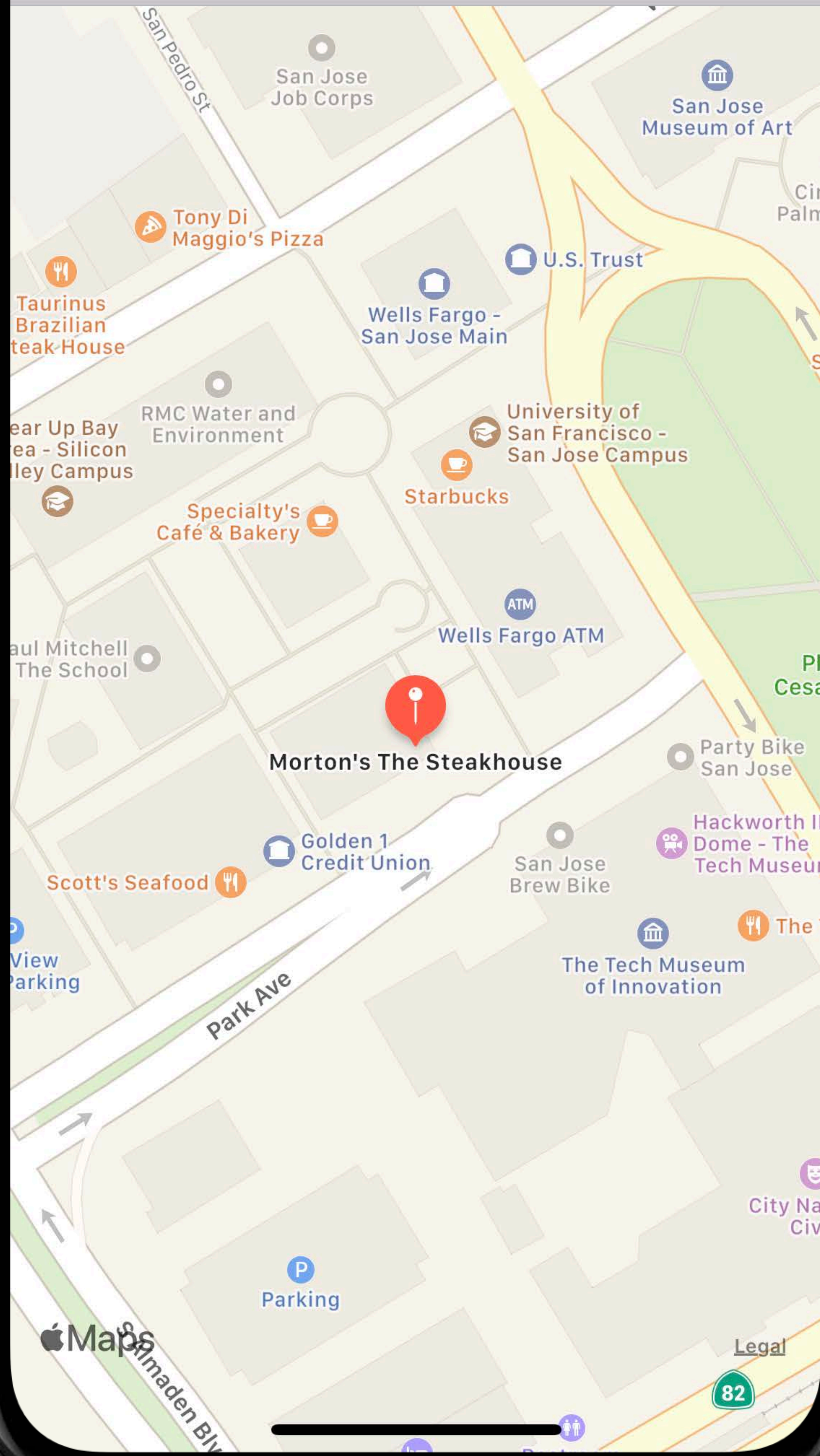
< Back

After Hours

🔍 Morton



Cancel



Apple Maps

Legal

9:41



< Back

After Hours

Q Mor|



Cancel

Morgan Hill, CA, United States

Morton's The Steakhouse

177 Park Ave, San Jose, CA 95113, United States

Morris Dailey Auditorium - San Jose State Uni...

1 Washington Sq, San Jose, CA 95192, United States

Morphe

2855 Stevens Creek Blvd, Santa Clara, CA 95050-6709, Unite...

Morrill Middle School

1970 Morrill Ave, San Jose, CA 95132, United States

Morrison Park Dr

San Jose, CA, United States

Morse St

San Jose, CA, United States

q w e r t y u i o p

a s d f g h j k l

⬆ z x c v b n m ⬇

123

space

Search



9:41



< Back

After Hours

Q Mor|



Cancel

Morgan Hill, CA, United States

Morton's The Steakhouse

177 Park Ave, San Jose, CA 95113, United States

Morris Dailey Auditorium - San Jose State Uni...

1 Washington Sq, San Jose, CA 95192, United States

Morphe

2855 Stevens Creek Blvd, Santa Clara, CA 95050-6709, Unite...

Morrill Middle School

1970 Morrill Ave, San Jose, CA 95132, United States

Morrison Park Dr

San Jose, CA, United States

Morse St

San Jose, CA, United States

q w e r t y u i o p

a s d f g h j k l

⬆ z x c v b n m ⬇

123

space

Search




```
// Create point of interest filter
let categories = [ .restaurant, .nightlife ]
let pointOfInterestFilter = MKPointOfInterestFilter(including: categories)

// Autocompletion Setup
let completer = MKLocalSearchCompleter()
completer.pointOfInterestFilter = pointOfInterestFilter
...
completer.queryFragment = query

// Search Request Setup
let request = MKLocalSearch.Request()
request.naturalLanguageQuery = query
request.pointOfInterestFilter = pointOfInterestFilter
...
request.start(completionHandler: completionHandler)
```



```
// Create point of interest filter
let categories = [ .restaurant, .nightlife ]
let pointOfInterestFilter = MKPointOfInterestFilter(including: categories)

// Autocompletion Setup
let completer = MKLocalSearchCompleter()
completer.pointOfInterestFilter = pointOfInterestFilter
...
completer.queryFragment = query

// Search Request Setup
let request = MKLocalSearch.Request()
request.naturalLanguageQuery = query
request.pointOfInterestFilter = pointOfInterestFilter
...
request.start(completionHandler: completionHandler)
```



```
// Create point of interest filter
let categories = [ .restaurant, .nightlife ]
let pointOfInterestFilter = MKPointOfInterestFilter(including: categories)

// Autocompletion Setup
let completer = MKLocalSearchCompleter()
completer.pointOfInterestFilter = pointOfInterestFilter
...
completer.queryFragment = query

// Search Request Setup
let request = MKLocalSearch.Request()
request.naturalLanguageQuery = query
request.pointOfInterestFilter = pointOfInterestFilter
...
request.start(completionHandler: completionHandler)
```



```
// Create point of interest filter
let categories = [ .restaurant, .nightlife ]
let pointOfInterestFilter = MKPointOfInterestFilter(including: categories)

// Autocompletion Setup
let completer = MKLocalSearchCompleter()
completer.pointOfInterestFilter = pointOfInterestFilter
...
completer.queryFragment = query

// Search Request Setup
let request = MKLocalSearch.Request()
request.naturalLanguageQuery = query
request.pointOfInterestFilter = pointOfInterestFilter
...
request.start(completionHandler: completionHandler)
```


9:41



< Back

After Hours

Q Mor|



Cancel

Morgan Hill, CA, United States

Morton's The Steakhouse

177 Park Ave, San Jose, CA 95113, United States

Morrison Park Dr

San Jose, CA, United States

Morse St

San Jose, CA, United States

Morse St

Santa Clara, CA, United States

q w e r t y u i o p

a s d f g h j k l

⬆ z x c v b n m ⬇

123

space

Search



9:41



< Back

After Hours



Mor|



Cancel

Morgan Hill, CA, United States

Morton's The Steakhouse

177 Park Ave, San Jose, CA 95113, United States

Morrison Park Dr

San Jose, CA, United States

Morse St

San Jose, CA, United States

Morse St

Santa Clara, CA, United States

q w e r t y u i o p

a s d f g h j k l

⬆ z x c v b n m ⬇

123

space

Search




```
// Autocompletion Setup
let completer = MKLocalSearchCompleter()
completer.delegate = self
completer.pointOfInterestFilter = pointOfInterestFilter

completer.filterType = .locationsOnly

...

// User input has changed
completer.queryFragment = query
```



```
// Autocompletion Setup
let completer = MKLocalSearchCompleter()
completer.delegate = self
completer.pointOfInterestFilter = pointOfInterestFilter
```

```
completer.filterType = .locationsOnly
```

```
...
```

```
// User input has changed
completer.queryFragment = query
```


ResultType

MKLocalSearchCompleter



NEW

ResultType

MKLocalSearchCompleter



NEW

.address

Example: 10600 N Tantau Ave, Cupertino, CA 95014

.pointOfInterest

Example: Apple Park Visitor Center

.queries

Example: coffee

ResultType

MKLocalSearch.Request



NEW

.address

Example: 10600 N Tantau Ave, Cupertino, CA 95014

.pointOfInterest

Example: Apple Park Visitor Center

Result Type Filtering

```
// Autocompletion Setup
let completer = MKLocalSearchCompleter()
completer.delegate = self
completer.pointOfInterestFilter = pointOfInterestFilter

completer.resultTypes = .pointOfInterest
```

```
// Search Request Setup
let request = MKLocalSearch.Request()
request.naturalLanguageQuery = query
request.pointOfInterestFilter = pointOfInterestFilter

request.resultTypes = .pointOfInterest
```


Result Type Filtering

```
// Autocompletion Setup
let completer = MKLocalSearchCompleter()
completer.delegate = self
completer.pointOfInterestFilter = pointOfInterestFilter

completer.resultTypes = .pointOfInterest
```

```
// Search Request Setup
let request = MKLocalSearch.Request()
request.naturalLanguageQuery = query
request.pointOfInterestFilter = pointOfInterestFilter

request.resultTypes = .pointOfInterest
```


Result Type Filtering

```
// Autocompletion Setup
let completer = MKLocalSearchCompleter()
completer.delegate = self
completer.pointOfInterestFilter = pointOfInterestFilter

completer.resultTypes = .pointOfInterest
```

```
// Search Request Setup
let request = MKLocalSearch.Request()
request.naturalLanguageQuery = query
request.pointOfInterestFilter = pointOfInterestFilter

request.resultTypes = .pointOfInterest
```


MKMapItem

MKMapItem

Location

Address

Name

Phone number

URL

MKMapItem

NEW

Location

Address

Name

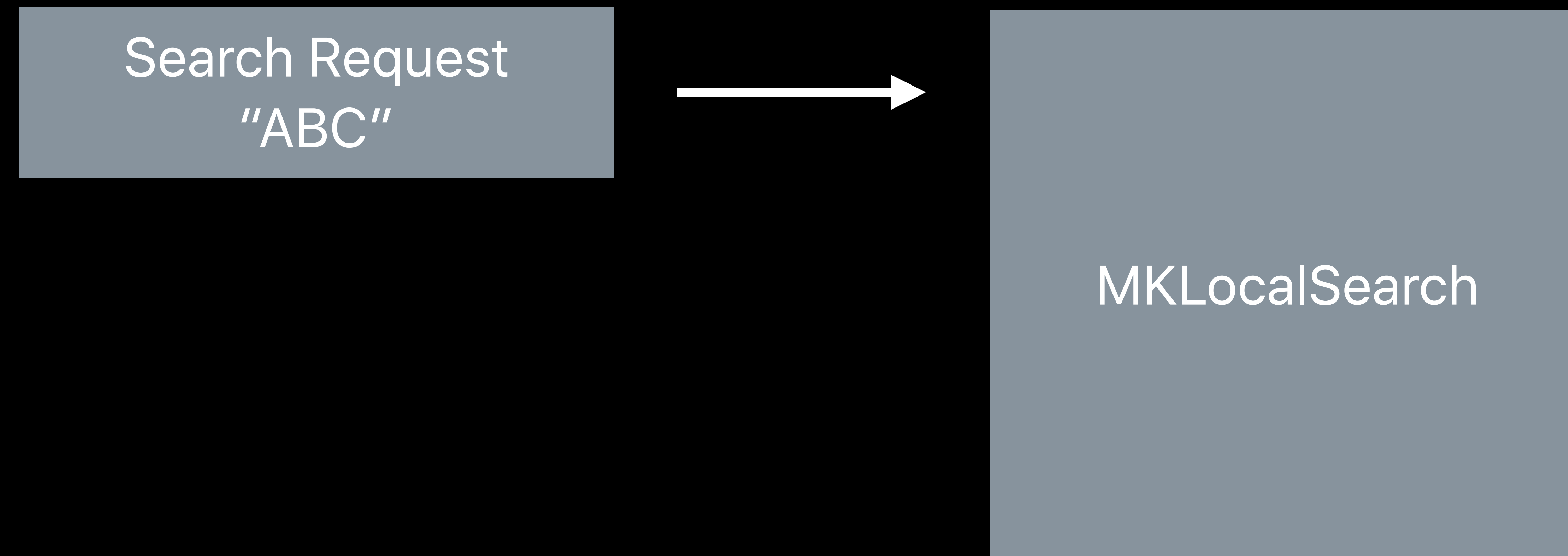
Phone number

URL

Point of interest category

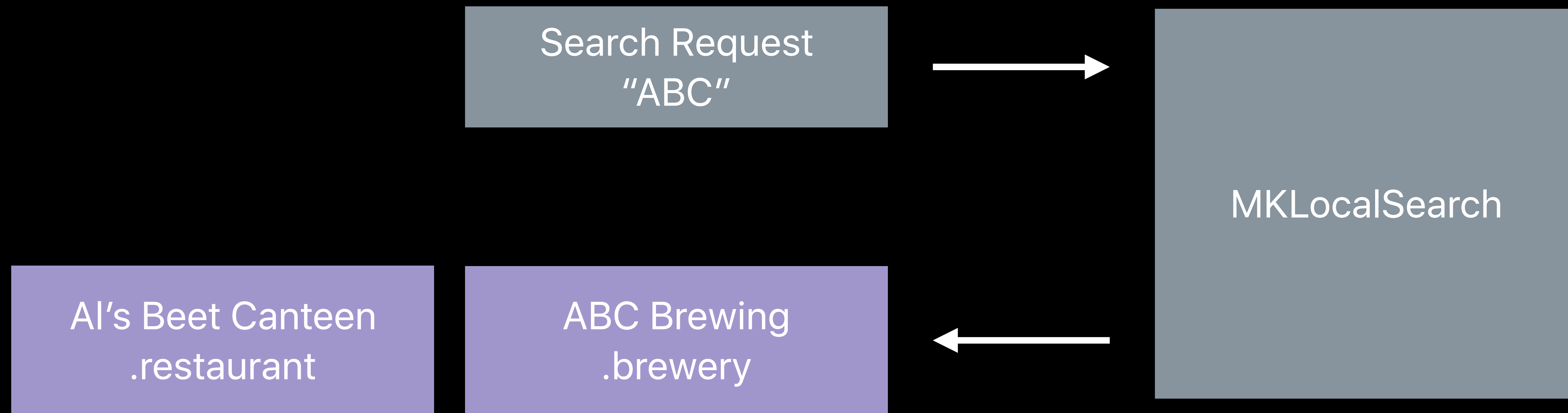
MKMapItem

Point of interest category



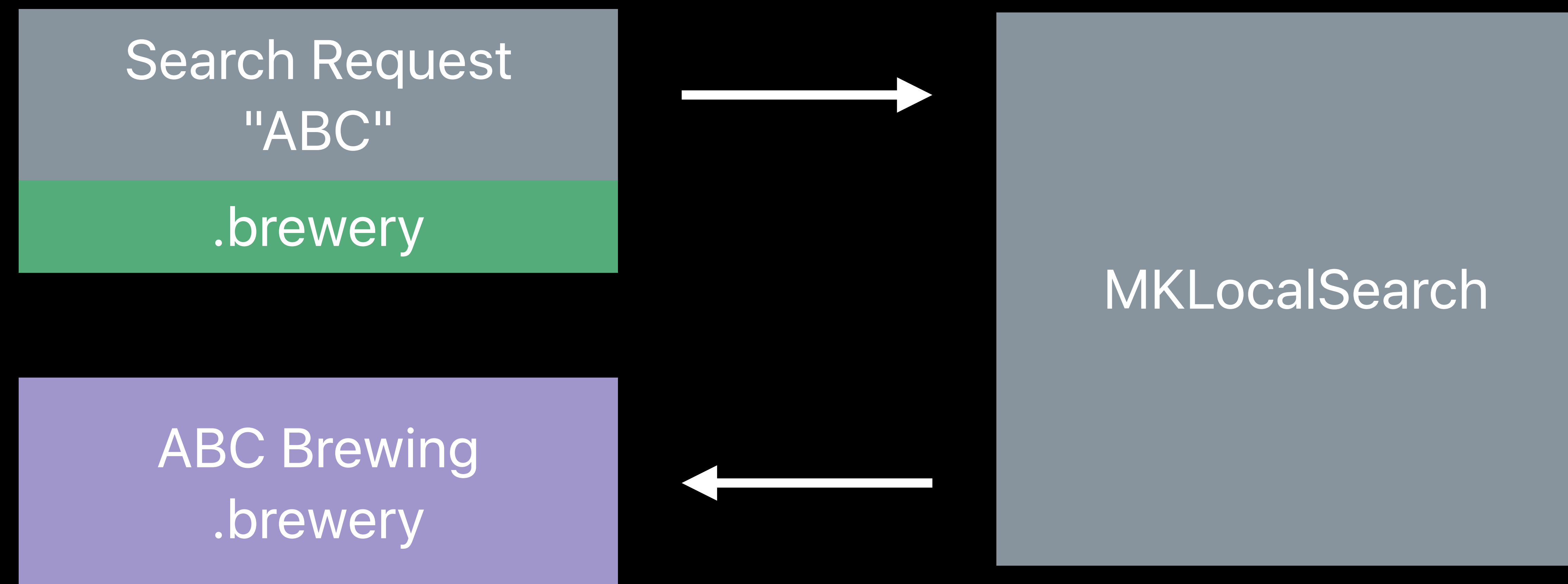
MKMapItem

Point of interest category



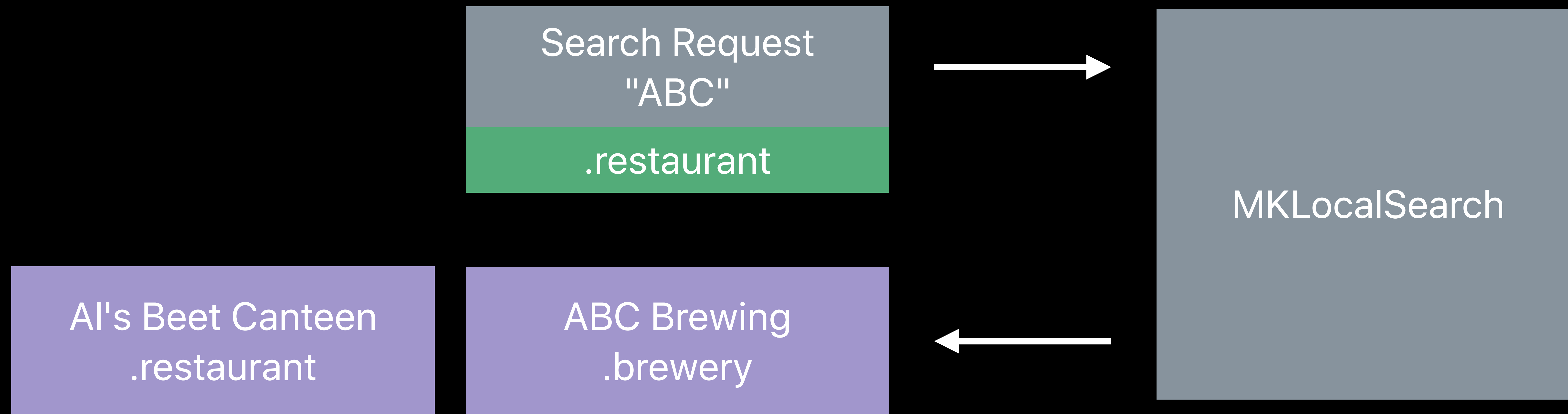
MKMapItem

Point of interest category



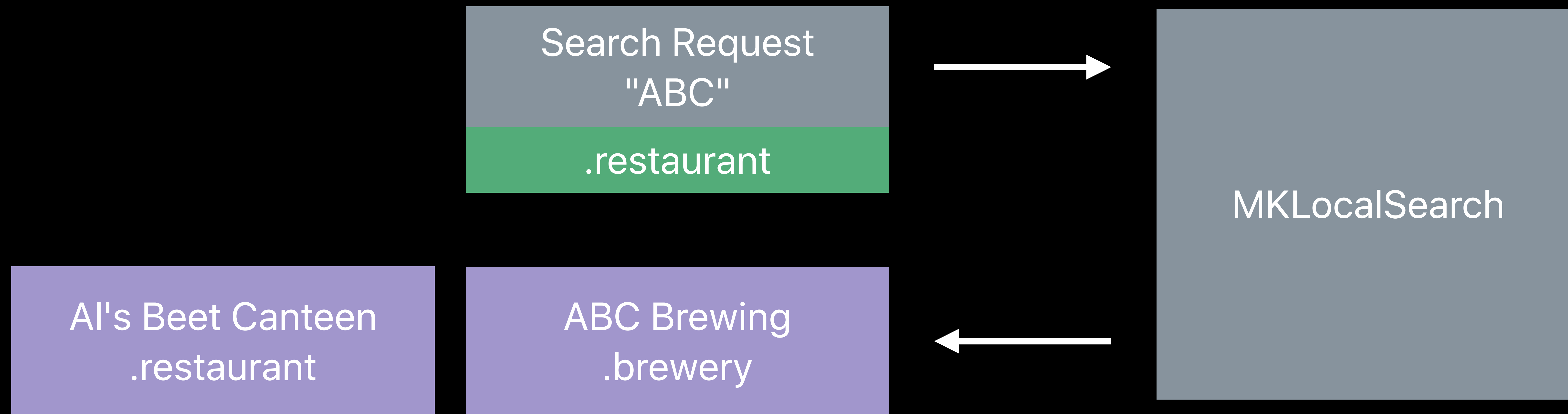
MKMapItem

Point of interest category



MKMapItem

Point of interest category



Search and Autocomplete Filtering in MapKit JS

NEW

Coming this fall!

```
// Filtering points of interest
let filter = mapkit.PointOfInterestFilter.including([ mapkit.PointOfInterestCategory.Hotel ]);
search.pointOfInterestFilter = filter;

// Filtering result types
search.includeAddresses = false;
search.includePointsOfInterest = true;
search.includeQueries = false;
```


Demo

Improving search and completion results

Nalini Shah, MapKit Engineer

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

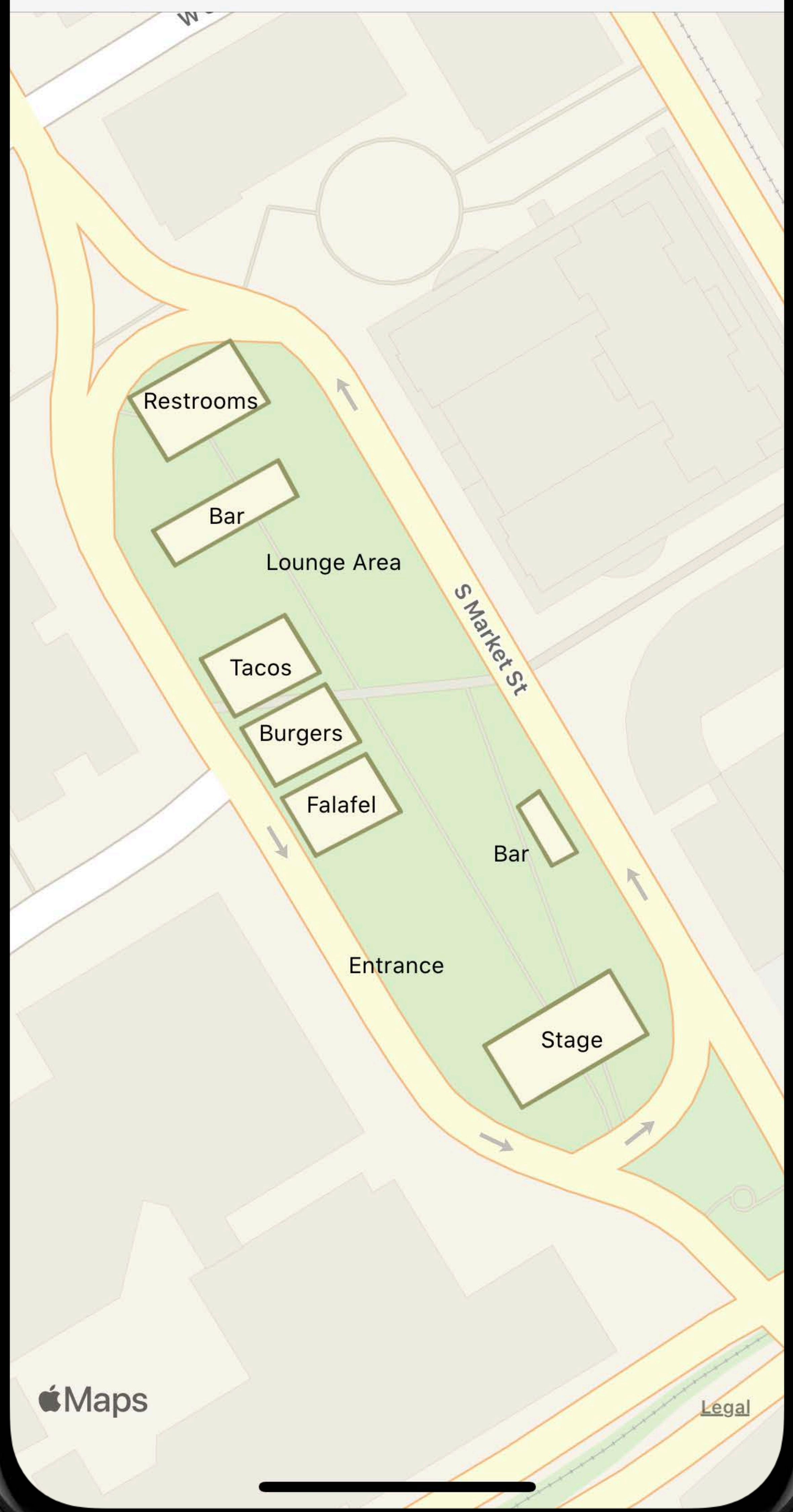
Taking control of the map view camera

9:41



[Back](#)

Concert in The Park



Apple Maps

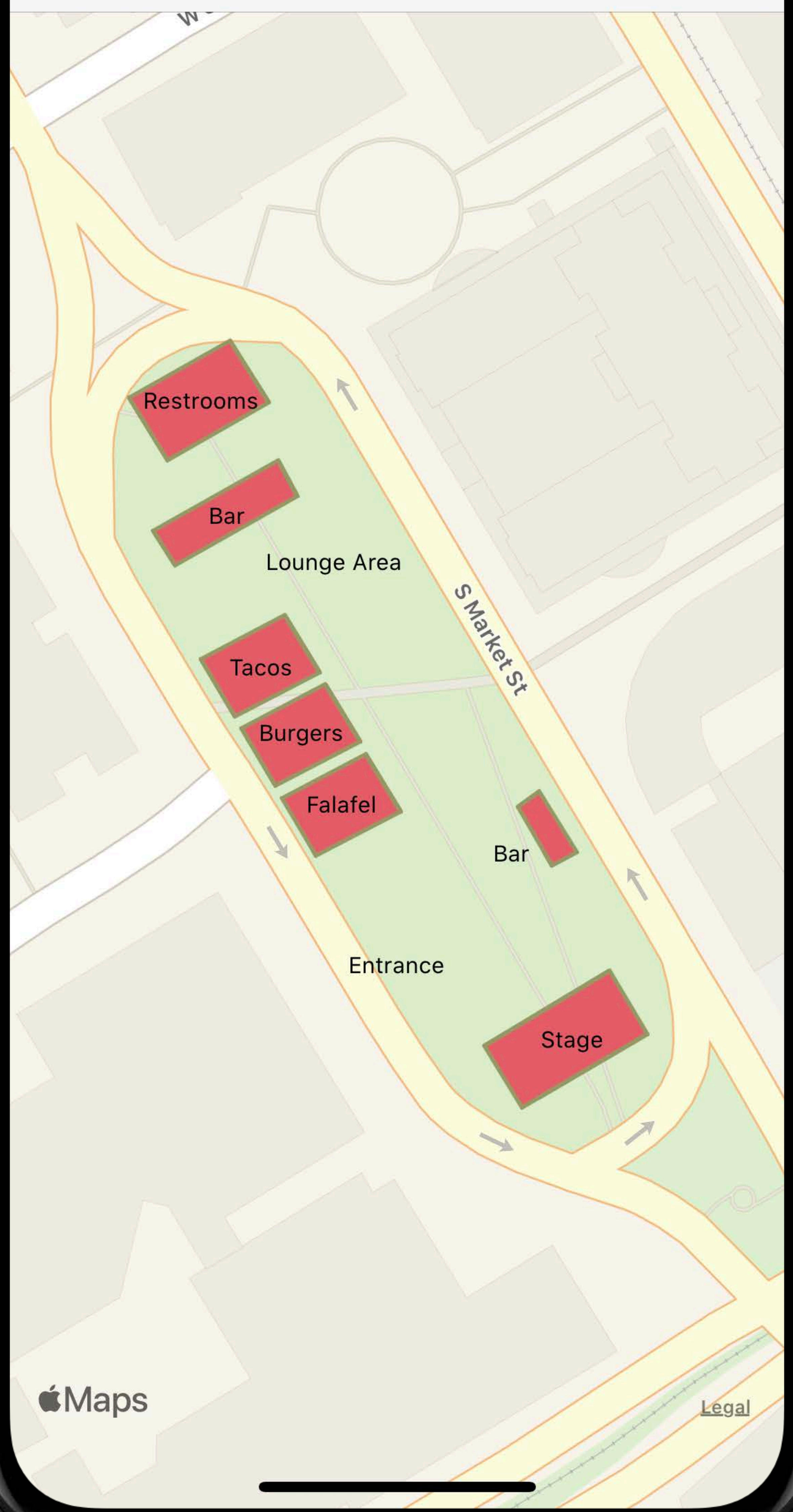
Legal

9:41



[← Back](#)

Concert in The Park



Apple Maps

Legal

Improving Polygon and Polyline Performance

NEW

Improving Polygon and Polyline Performance

NEW

MKMultiPolygon

- Groups polygons of the same style

MKMultiPolyline

- Groups polylines of the same style

Improving Polygon and Polyline Performance

NEW

Improving Polygon and Polyline Performance



NEW

MKMultiPolygonRenderer

- Renders a group of polygons of the same style

MKMultiPolylineRenderer

- Renders a group of polylines of the same style


```
// Adding polygons to a map view

let stageCoordinates = [
    CLLocationCoordinate2D(latitude: 37.332536, longitude: -121.889562),
    CLLocationCoordinate2D(latitude: 37.332312, longitude: -121.890037),
    CLLocationCoordinate2D(latitude: 37.332082, longitude: -121.889863),
    CLLocationCoordinate2D(latitude: 37.332310, longitude: -121.889393)
]

let stage = MKPolygon(coordinates: stageCoordinates, count: 4)

// Creation of other polygons omitted...

mapView.addOverlays([stage, mainBar, bar, tacos, burgers, falafel, restroom])
```



```
// Adding polygons to a map view
```

```
let stageCoordinates = [  
    CLLocationCoordinate2D(latitude: 37.332536, longitude: -121.889562),  
    CLLocationCoordinate2D(latitude: 37.332312, longitude: -121.890037),  
    CLLocationCoordinate2D(latitude: 37.332082, longitude: -121.889863),  
    CLLocationCoordinate2D(latitude: 37.332310, longitude: -121.889393)  
]
```

```
let stage = MKPolygon(coordinates: stageCoordinates, count: 4)
```

```
// Creation of other polygons omitted...
```

```
mapView.addOverlays([stage, mainBar, bar, tacos, burgers, falafel, restroom])
```



```
// Adding polygons to a map view

let stageCoordinates = [
  CLLocationCoordinate2D(latitude: 37.332536, longitude: -121.889562),
  CLLocationCoordinate2D(latitude: 37.332312, longitude: -121.890037),
  CLLocationCoordinate2D(latitude: 37.332082, longitude: -121.889863),
  CLLocationCoordinate2D(latitude: 37.332310, longitude: -121.889393)
]

let stage = MKPolygon(coordinates: stageCoordinates, count: 4)

// Creation of other polygons omitted...

mapView.addOverlays([stage, mainBar, bar, tacos, burgers, falafel, restroom])
```



```
// Adding a multipolygon to a map view

let stageCoordinates = [
    CLLocationCoordinate2D(latitude: 37.332536, longitude: -121.889562),
    CLLocationCoordinate2D(latitude: 37.332312, longitude: -121.890037),
    CLLocationCoordinate2D(latitude: 37.332082, longitude: -121.889863),
    CLLocationCoordinate2D(latitude: 37.332310, longitude: -121.889393)
]

let stage = MKPolygon(coordinates: stageCoordinates, count: 4)

// Creation of other polygons omitted...

let venue = MKMultiPolygon(polygons: [stage, mainBar, bar, tacos, burgers, falafel, restroom])
mapView.addOverlay(venue)
```



```
// Adding a multipolygon to a map view
```

```
let stageCoordinates = [  
    CLLocationCoordinate2D(latitude: 37.332536, longitude: -121.889562),  
    CLLocationCoordinate2D(latitude: 37.332312, longitude: -121.890037),  
    CLLocationCoordinate2D(latitude: 37.332082, longitude: -121.889863),  
    CLLocationCoordinate2D(latitude: 37.332310, longitude: -121.889393)  
]
```

```
let stage = MKPolygon(coordinates: stageCoordinates, count: 4)
```

```
// Creation of other polygons omitted...
```

```
let venue = MKMultiPolygon(polygons: [stage, mainBar, bar, tacos, burgers, falafel, restroom])  
mapView.addOverlay(venue)
```



```
// Adding a multipolygon to a map view

let stageCoordinates = [
    CLLocationCoordinate2D(latitude: 37.332536, longitude: -121.889562),
    CLLocationCoordinate2D(latitude: 37.332312, longitude: -121.890037),
    CLLocationCoordinate2D(latitude: 37.332082, longitude: -121.889863),
    CLLocationCoordinate2D(latitude: 37.332310, longitude: -121.889393)
]

let stage = MKPolygon(coordinates: stageCoordinates, count: 4)

// Creation of other polygons omitted...

let venue = MKMultiPolygon(polygons: [stage, mainBar, bar, tacos, burgers, falafel, restroom])
mapView.addOverlay(venue)
```



```
// Providing a multipolygon renderer

func mapView(_ mapView: MKMapView, rendererFor overlay: MKOverlay) -> MKOverlayRenderer {

    if let multiPolygon = overlay as? MKMultiPolygon {
        let renderer = MKMultiPolygonRenderer(polygon: multiPolygon)
        renderer.fillColor = UIColor(named: "VenueFill")
        renderer.strokeColor = UIColor(named: "VenueStroke")
        renderer.lineWidth = 2.0

        return renderer
    }

    return MKOverlayRenderer(overlay: overlay)
}
```



```
// Providing a multipolygon renderer

func mapView(_ mapView: MKMapView, rendererFor overlay: MKOverlay) -> MKOverlayRenderer {

    if let multiPolygon = overlay as? MKMultiPolygon {
        let renderer = MKMultiPolygonRenderer(polygon: multiPolygon)
        renderer.fillColor = UIColor(named: "VenueFill")
        renderer.strokeColor = UIColor(named: "VenueStroke")
        renderer.lineWidth = 2.0

        return renderer
    }

    return MKOverlayRenderer(overlay: overlay)
}
```



```
// Providing a multipolygon renderer

func mapView(_ mapView: MKMapView, rendererFor overlay: MKOverlay) -> MKOverlayRenderer {

    if let multiPolygon = overlay as? MKMultiPolygon {
        let renderer = MKMultiPolygonRenderer(polygon: multiPolygon)
        renderer.fillColor = UIColor(named: "VenueFill")
        renderer.strokeColor = UIColor(named: "VenueStroke")
        renderer.lineWidth = 2.0

        return renderer
    }

    return MKOverlayRenderer(overlay: overlay)
}
```



```
// Opt out of vector rendering
```

```
open class MKOverlayPathRenderer: MKOverlayRenderer {
```

```
    open var shouldRasterize: Bool
```

```
}
```



NEW



NEW

```
// Opt out of vector rendering
```

```
open class MKOverlayPathRenderer: MKOverlayRenderer {
```

```
    open var shouldRasterize: Bool
```

```
}
```


Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera


```
// Example GeoJSON (RFC 7946)
```

```
{  
  "type": "Feature",  
  "id": "SampleFeatureID",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [  
      [  
        [ -121.890354, 37.332744 ]  
      ]  
    ]  
  },  
  "properties": {  
    "name": "Stage"  
  }  
}
```



```
// Example GeoJSON (RFC 7946)
```

```
{  
  "type": "Feature",  
  "identifier": "SampleFeatureID",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [  
      [  
        [ -121.890354, 37.332744 ]  
      ]  
    ]  
  },  
  "properties": {  
    "name": "Stage"  
  }  
}
```



```
// Example GeoJSON (RFC 7946)
```

```
{  
  "type": "Feature",  
  "id": "SampleFeatureID",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [  
      [  
        [ -121.890354, 37.332744 ]  
      ]  
    ]  
  },  
  "properties": {  
    "name": "Stage"  
  }  
}
```


MKGeoJSONFeature

NEW

MKGeoJSONFeature

NEW

Identifier

Geometry

Properties data

MKGeoJSONDecoder

Decoding GeoJSON data into MapKit objects



NEW

MKGeoJSONDecoder

Decoding GeoJSON data into MapKit objects

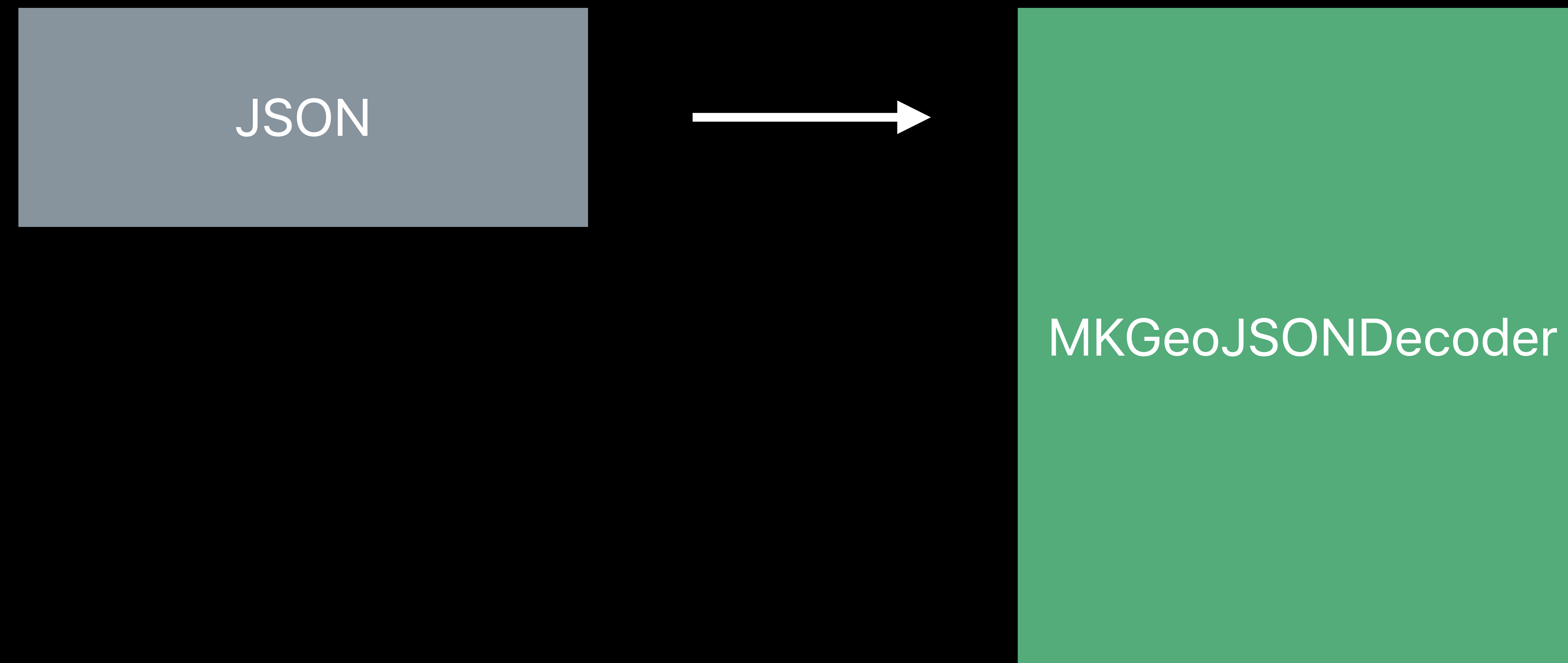
NEW

MKGeoJSONDecoder

MKGeoJSONDecoder

Decoding GeoJSON data into MapKit objects

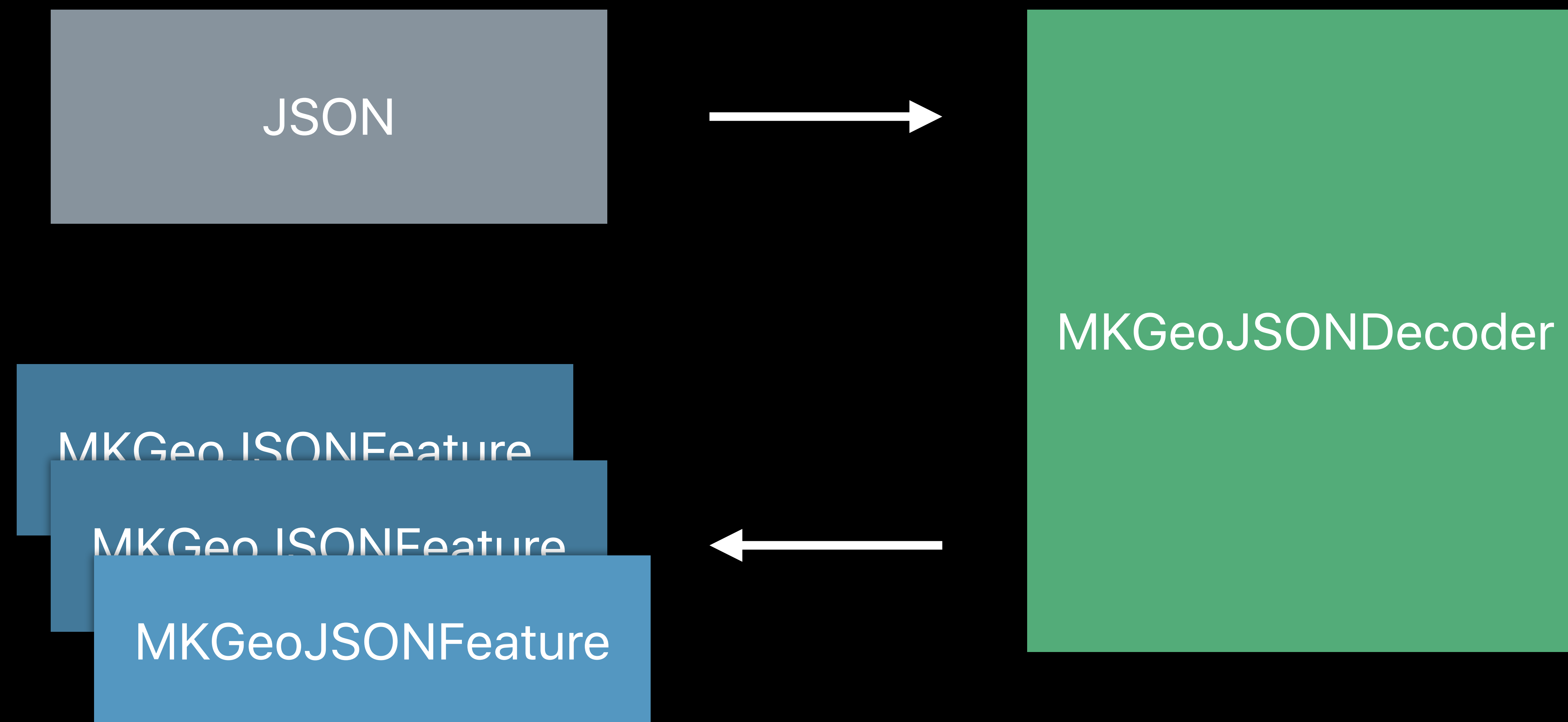
NEW



MKGeoJSONDecoder

Decoding GeoJSON data into MapKit objects

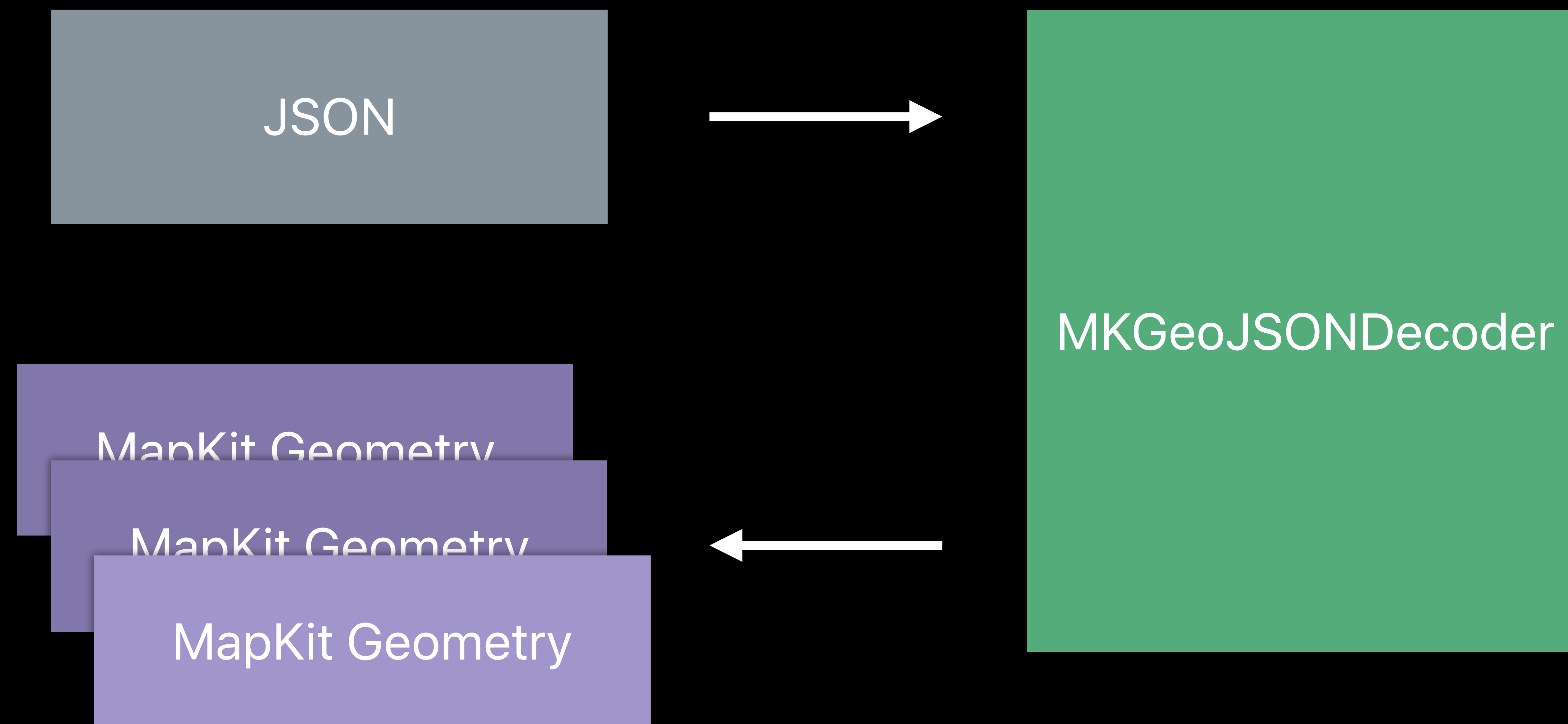
NEW



MKGeoJSONDecoder

Decoding GeoJSON data into MapKit objects

NEW




```
// Example GeoJSON (RFC 7946)

{
  "type": "Feature",
  "id": "SampleFeatureID",
  "geometry": {
    "type": "Point",
    "coordinates": [
      [
        [ -121.890354, 37.332744 ]
      ]
    ]
  },
  "properties": {
    "name": "Stage"
  }
}
```

A green rectangular box containing the text "MKGeoJSONDecoder" in white, sans-serif font, centered within the box.

MKGeoJSONDecoder


```
// Example GeoJSON (RFC 7946)
```

```
{  
  "type": "Feature",  
  "id": "SampleFeatureID",  
  "geometry":
```

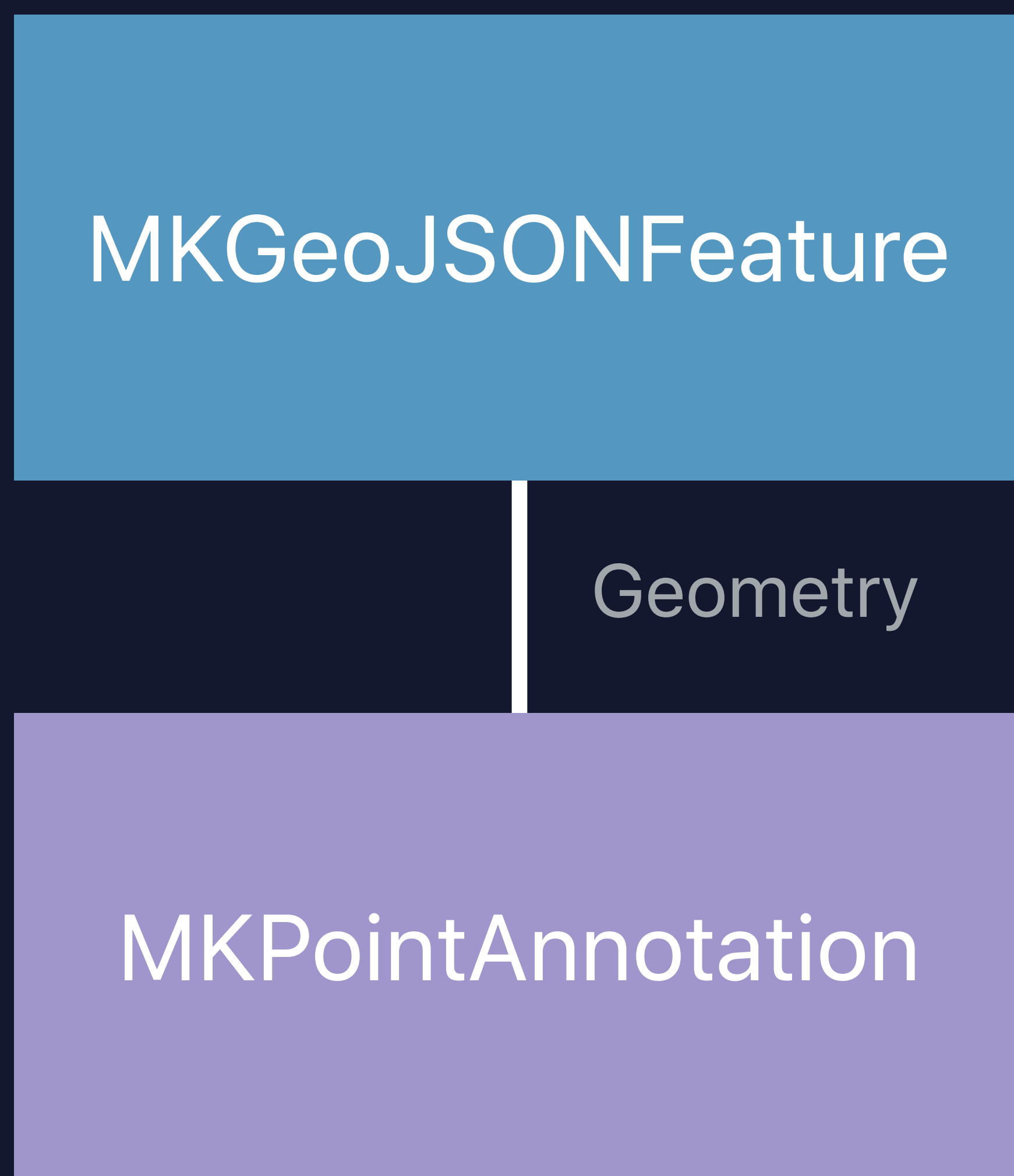
A light purple rectangular box containing the text "MKPointAnnotation".

MKPointAnnotation

A green rectangular box containing the text "MKGeoJSONDecoder".

MKGeoJSONDecoder

```
,  
  "properties": {  
    "name": "Stage"  
  }  
}
```

Mapping GeoJSON to MapKit

Mapping GeoJSON to MapKit

GeoJSON	MapKit
Point	MKPointAnnotation
MultiPoint	MKMultiPoint
LineString	MKPolyline
MultiLineString	MKMultiPolyline
Polygon	MKPolygon
MultiPolygon	MKMultiPolygon


```
// Decoding GeoJSON

do {
    let decoder = MKGeoJSONDecoder()
    let decodedObjects = try decoder.decode(venueData)

    for object in decodedObjects {
        if let feature = object as? MKGeoJSONFeature {
            parse(feature: feature)
        } else if let geometry = object as? (MKShape & MKGeoJSONObject) {
            parse(geometry: geometry)
        }
    }
} catch {
    print("Unexpected error: \(error).")
}
```



```
// Decoding GeoJSON

do {
    let decoder = MKGeoJSONDecoder()
    let decodedObjects = try decoder.decode(venueData)

    for object in decodedObjects {
        if let feature = object as? MKGeoJSONFeature {
            parse(feature: feature)
        } else if let geometry = object as? (MKShape & MKGeoJSONObject) {
            parse(geometry: geometry)
        }
    }
} catch {
    print("Unexpected error: \(error).")
}
```



```
// Decoding GeoJSON

do {
    let decoder = MKGeoJSONDecoder()
    let decodedObjects = try decoder.decode(venueData)

    for object in decodedObjects {
        if let feature = object as? MKGeoJSONFeature {
            parse(feature: feature)
        } else if let geometry = object as? (MKShape & MKGeoJSONObject) {
            parse(geometry: geometry)
        }
    }
} catch {
    print("Unexpected error: \(error).")
}
```



```
// Decoding GeoJSON

do {
    let decoder = MKGeoJSONDecoder()
    let decodedObjects = try decoder.decode(venueData)

    for object in decodedObjects {
        if let feature = object as? MKGeoJSONFeature {
            parse(feature: feature)
        } else if let geometry = object as? (MKShape & MKGeoJSONObject) {
            parse(geometry: geometry)
        }
    }
} catch {
    print("Unexpected error: \(error).")
}
```



```
// Decoding GeoJSON

do {
    let decoder = MKGeoJSONDecoder()
    let decodedObjects = try decoder.decode(venueData)

    for object in decodedObjects {
        if let feature = object as? MKGeoJSONFeature {
            parse(feature: feature)
        } else if let geometry = object as? (MKShape & MKGeoJSONObject) {
            parse(geometry: geometry)
        }
    }
} catch {
    print("Unexpected error: \(error).")
}
```



```
// Parsing a feature's 'properties' data

private func parse(feature: MKGeoJSONFeature) {

    if let point = feature.geometry.first as? MKPointAnnotation {

        if let properties = feature.properties {

            let decoder = JSONDecoder()
            if let dictionary = try? decoder.decode([String: String].self, from: properties) {
                point.title = dictionary["name"]
            }
        }

        annotations.append(point)
    }
}
```



```
// Parsing a feature's 'properties' data

private func parse(feature: MKGeoJSONFeature) {

    if let point = feature.geometry.first as? MKPointAnnotation {

        if let properties = feature.properties {

            let decoder = JSONDecoder()
            if let dictionary = try? decoder.decode([String: String].self, from: properties) {
                point.title = dictionary["name"]
            }
        }

        annotations.append(point)
    }
}
```



```
// Parsing a feature's 'properties' data

private func parse(feature: MKGeoJSONFeature) {

    if let point = feature.geometry.first as? MKPointAnnotation {

        if let properties = feature.properties {

            let decoder = JSONDecoder()
            if let dictionary = try? decoder.decode([String: String].self, from: properties) {
                point.title = dictionary["name"]
            }
        }

        annotations.append(point)
    }
}
```



```
// Parsing a feature's 'properties' data

private func parse(feature: MKGeoJSONFeature) {

    if let point = feature.geometry.first as? MKPointAnnotation {

        if let properties = feature.properties {

            let decoder = JSONDecoder()
            if let dictionary = try? decoder.decode([String: String].self, from: properties) {
                point.title = dictionary["name"]
            }
        }

        annotations.append(point)
    }
}
```



```
// Parsing a feature's 'properties' data

private func parse(feature: MKGeoJSONFeature) {

    if let point = feature.geometry.first as? MKPointAnnotation {

        if let properties = feature.properties {

            let decoder = JSONDecoder()
            if let dictionary = try? decoder.decode([String: String].self, from: properties) {
                point.title = dictionary["name"]
            }
        }

        annotations.append(point)
    }
}
```



```
// Parsing unknown 'properties' data

if let properties = feature.properties {

    if let json = try? JSONSerialization.jsonObject(with: properties, options: []) {

        if let dictionary = json as? [String: Any] {
            // Inspect Dictionary
        } else if let array = json as? [Any] {
            // Inspect array
        }
    }
}
}
```



```
// Parsing unknown 'properties' data

if let properties = feature.properties {

    if let json = try? JSONSerialization.jsonObject(with: properties, options: []) {

        if let dictionary = json as? [String: Any] {
            // Inspect Dictionary
        } else if let array = json as? [Any] {
            // Inspect array
        }
    }
}
}
```



```
// Parsing unknown 'properties' data

if let properties = feature.properties {

    if let json = try? JSONSerialization.jsonObject(with: properties, options: []) {

        if let dictionary = json as? [String: Any] {
            // Inspect Dictionary
        } else if let array = json as? [Any] {
            // Inspect array
        }

    }

}
```


Mapping GeoJSON to MapKit JS

Mapping GeoJSON to MapKit JS

GeoJSON	MapKit JS
Point	MarkerAnnotation
MultiPoint	ItemCollection
LineString	PolylineOverlay
MultiLineString	ItemCollection
Polygon	PolygonOverlay
MultiPolygon	ItemCollection
Feature	ItemCollection MarkerAnnotation Overlay


```
// Importing GeoJSON in MapKit JS

var mapKitItems = mapkit.importGeoJSON(
    "https://example.com/features.geojson",
    geoJsonDelegate
);

map.showItems(mapKitItems);
```


Demo

Decoding GeoJSON

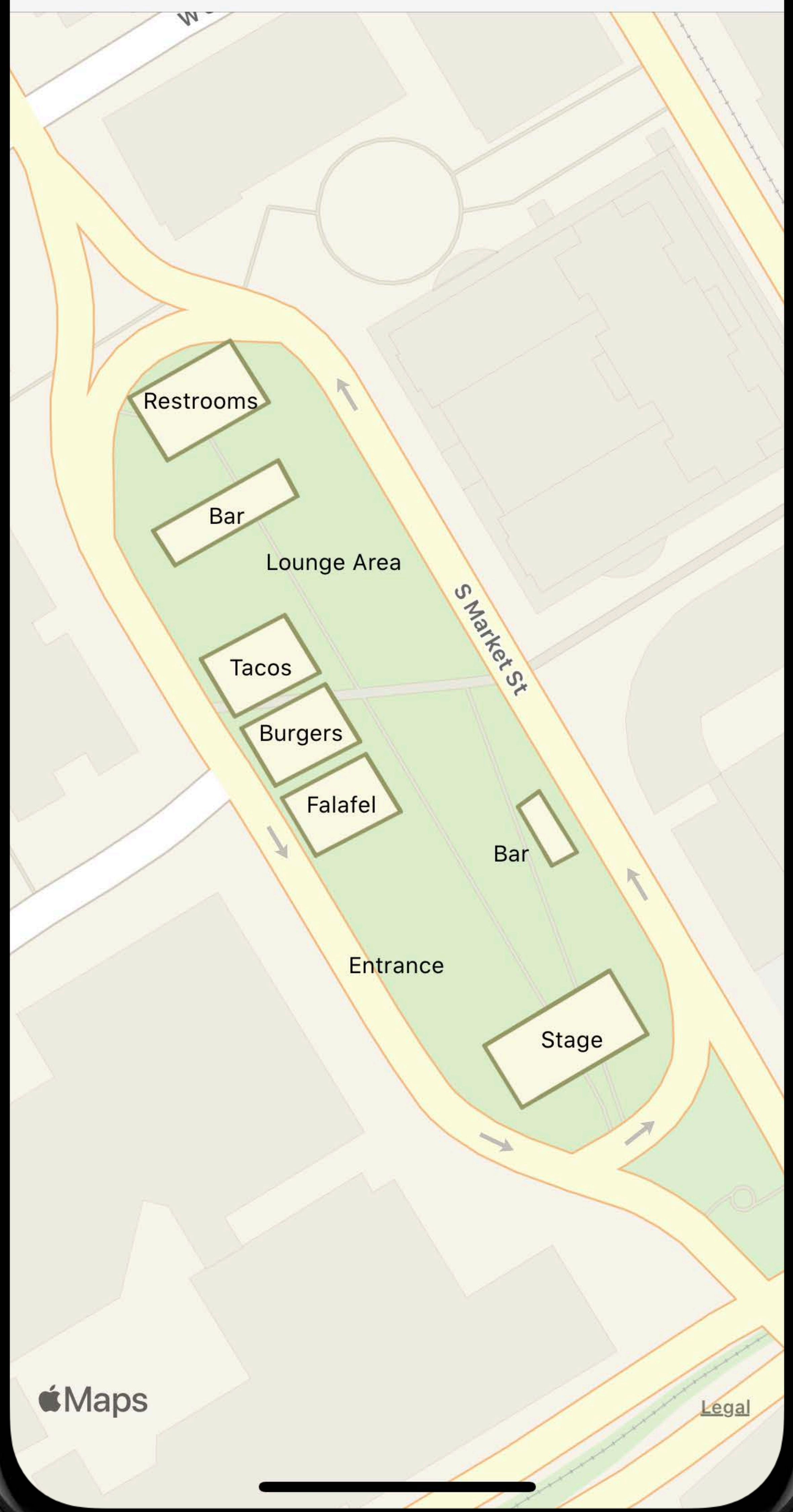
Nalini Shah, MapKit Engineer

9:41



[Back](#)

Concert in The Park



Apple Maps

Legal



Indoor Mapping Data Format

Indoor Mapping Data Format

Built on top of GeoJSON

Indoor Mapping Data Format

Built on top of GeoJSON

Complete set of features to represent a venue

Adding Indoor Maps to your App and Website

Friday 2:00

Introducing the Indoor Map Program

WWDC 2019

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

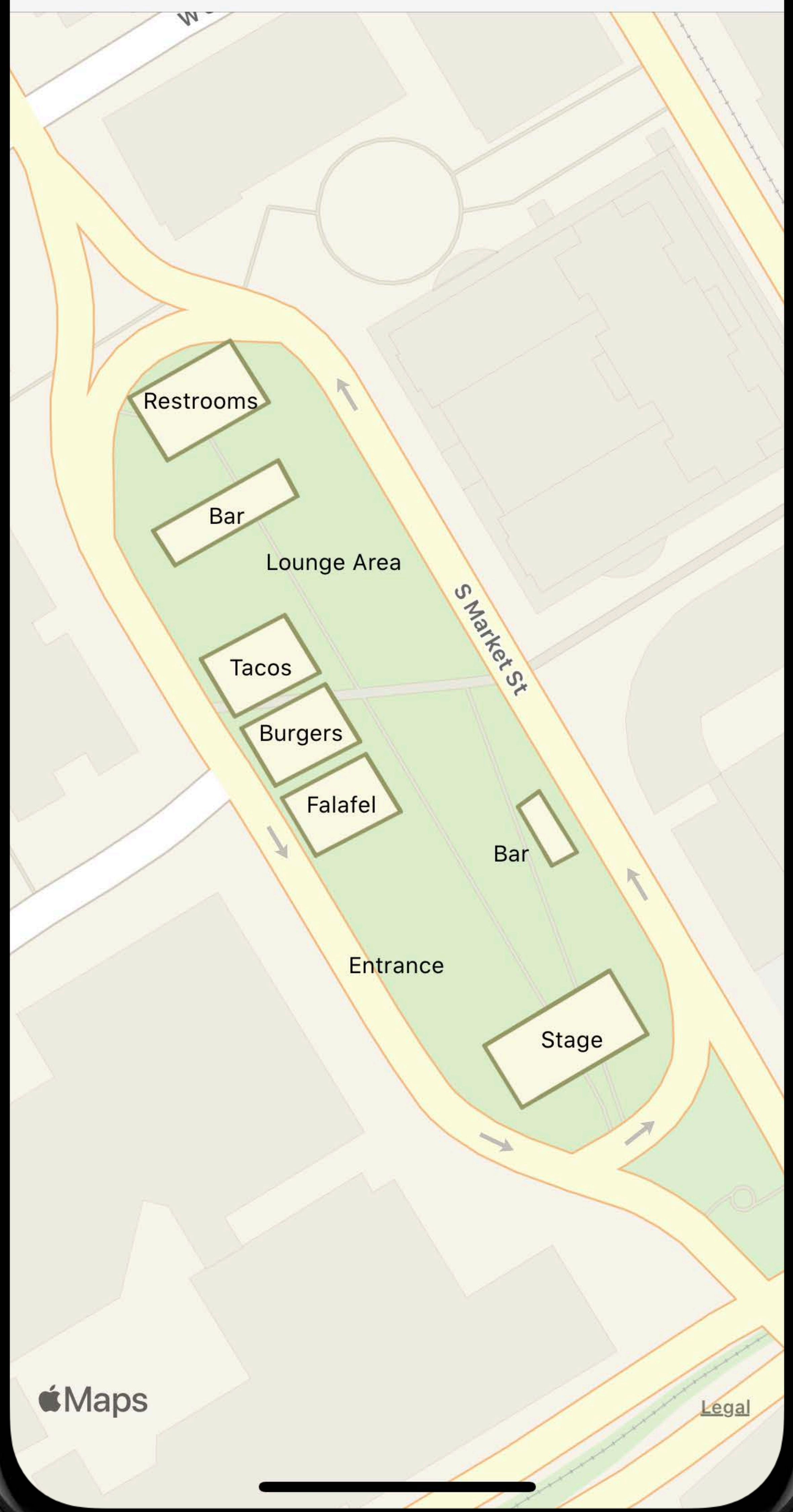
Taking control of the map view camera

9:41



[Back](#)

Concert in The Park



Apple Maps

Legal

9:41



[Back](#)

Concert in The Park



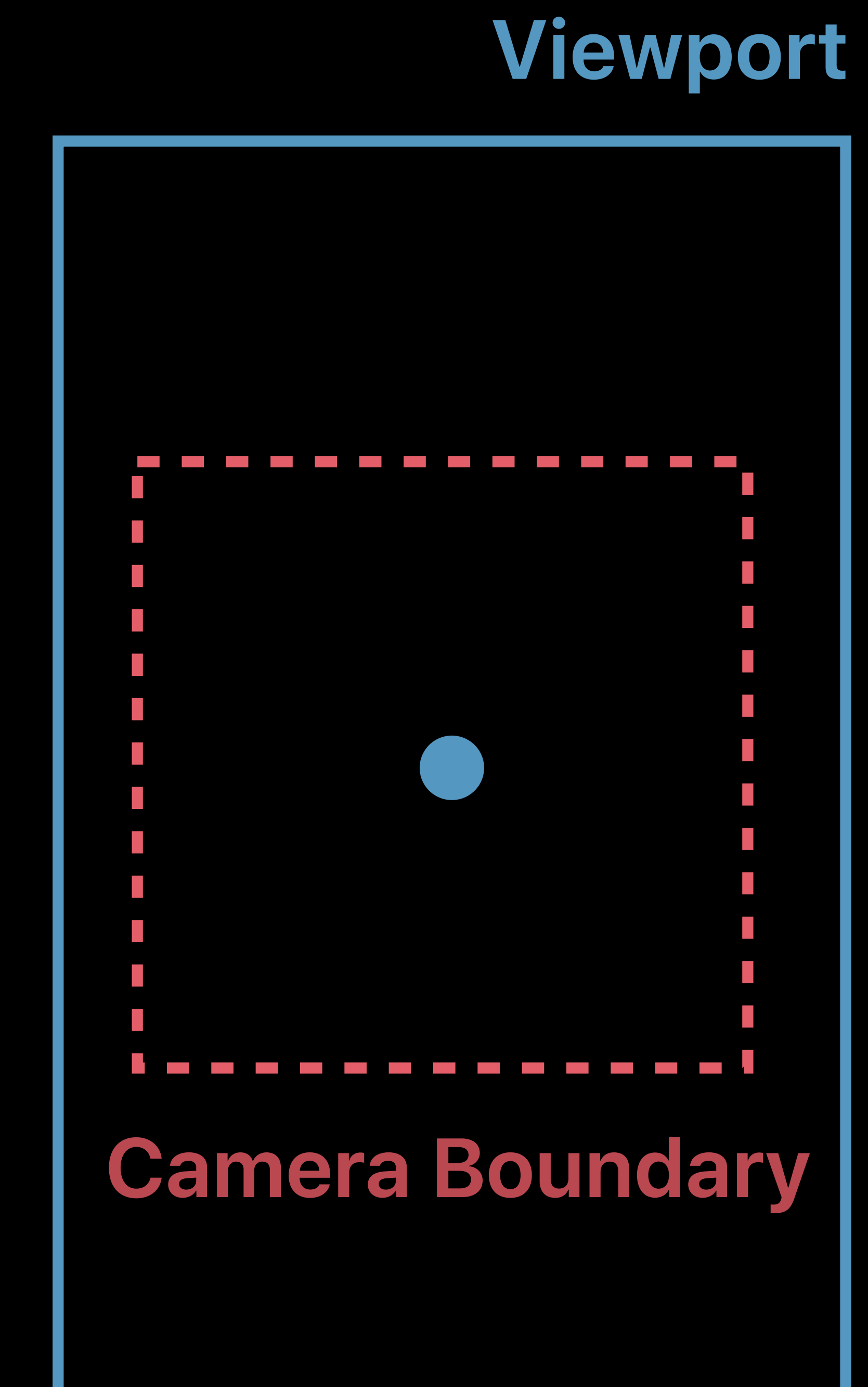
Apple Maps

Legal

MKMapView.CameraBoundary

NEW

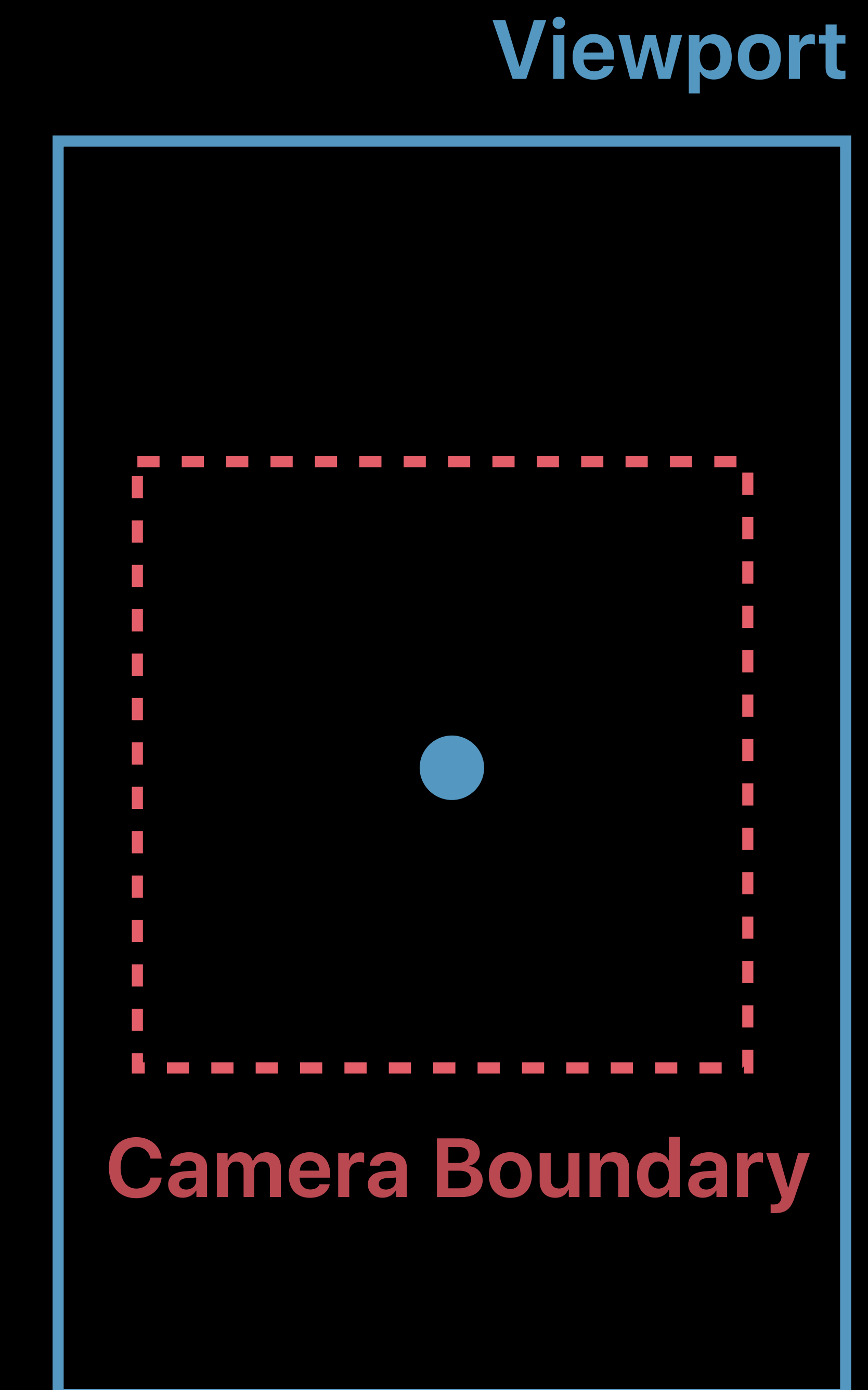
Constrains the map view's center point



MKMapView.CameraBoundary

NEW

Constrains the map view's center point

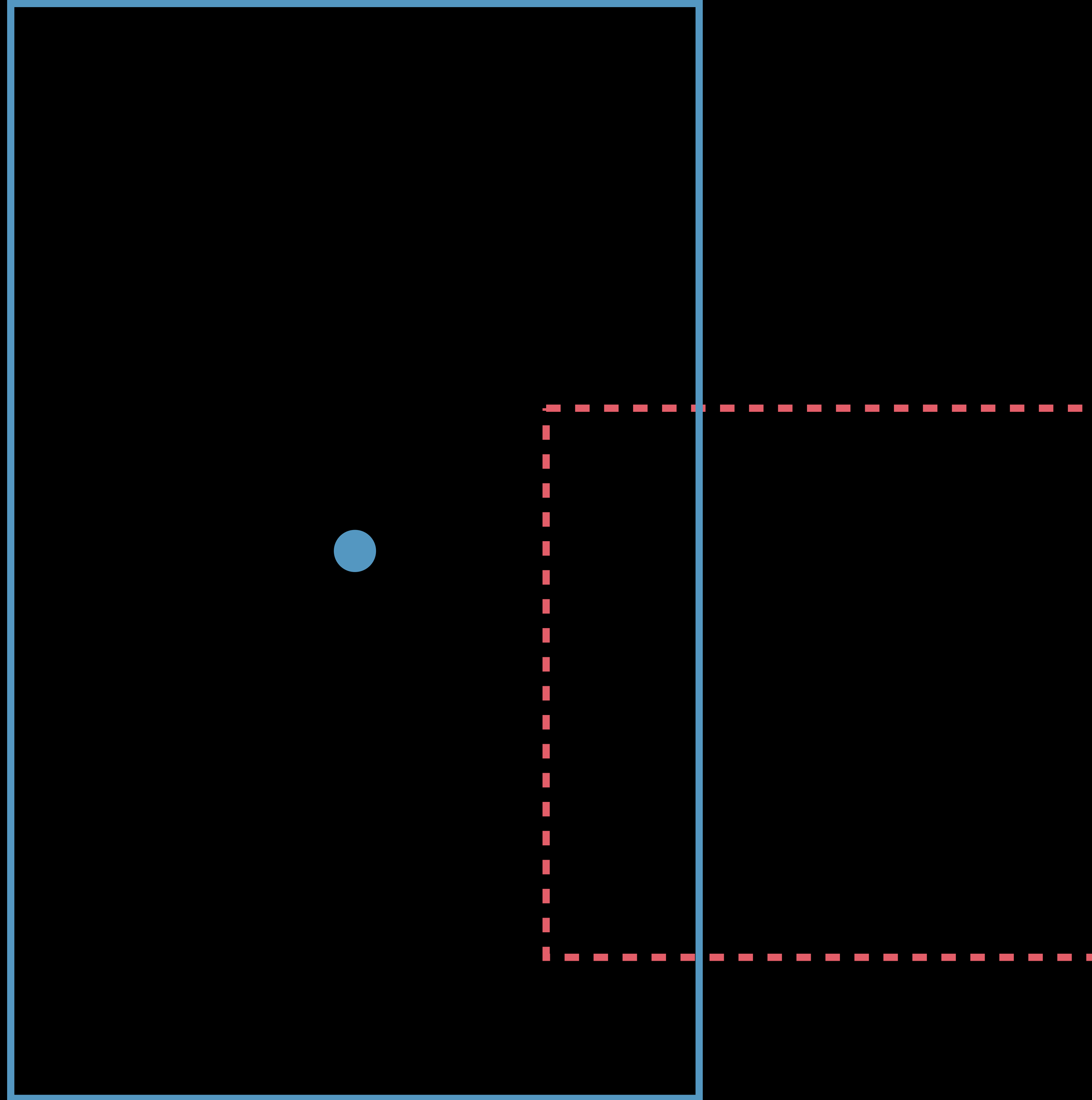


Applying a Camera Boundary

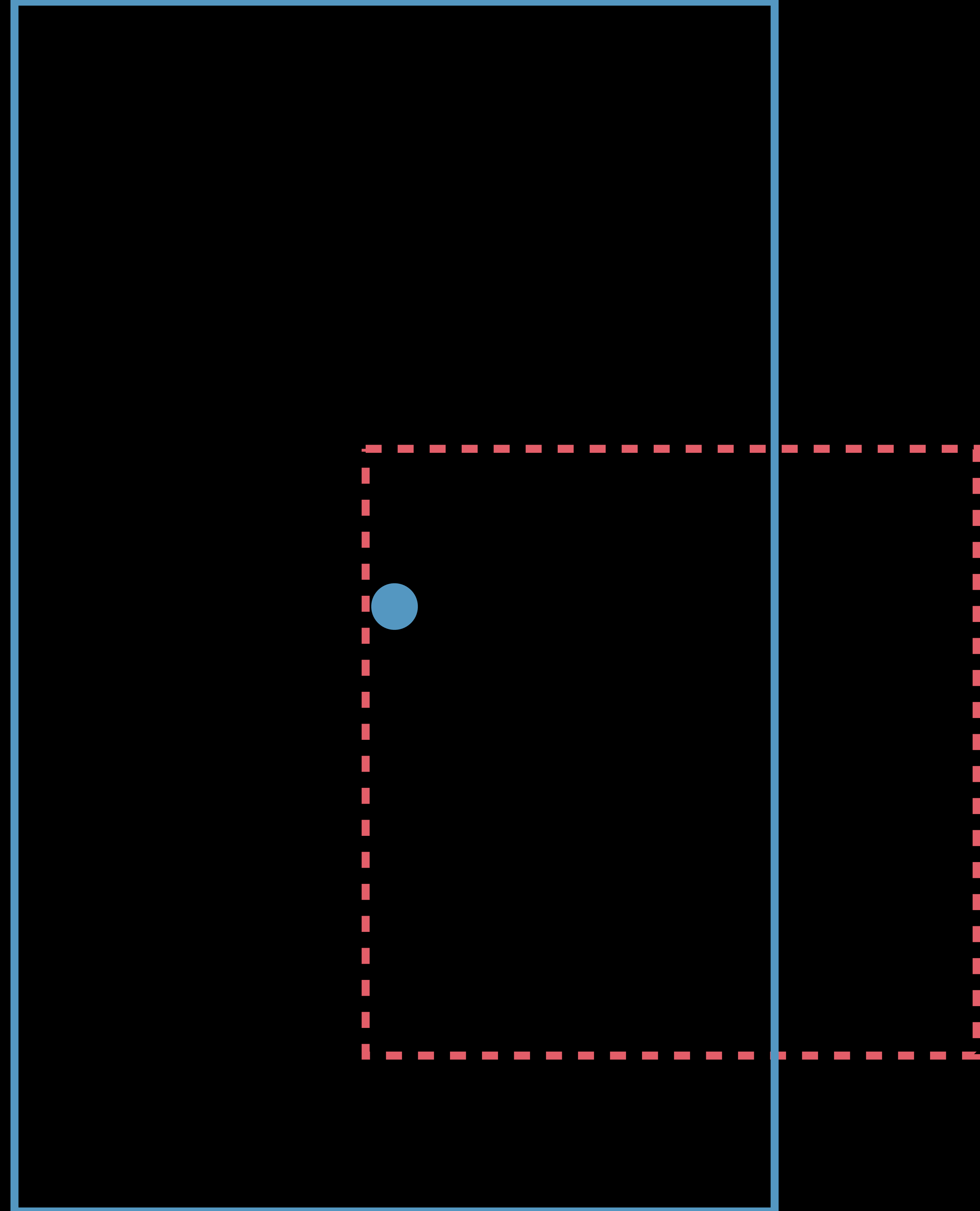
```
// Alternative 1: Create using an MKCoordinateRegion
let region = MKCoordinateRegion(center: center, latitudinalMeters: 80, longitudinalMeters: 90)
mapView.cameraBoundary = MKMapView.CameraBoundary(coordinateRegion: region)
```

```
// Alternative 2: Create using an MKMapRect
let mapRect = MKMapRect(origin: origin, size: size)
mapView.cameraBoundary = MKMapView.CameraBoundary(mapRect: mapRect)
```

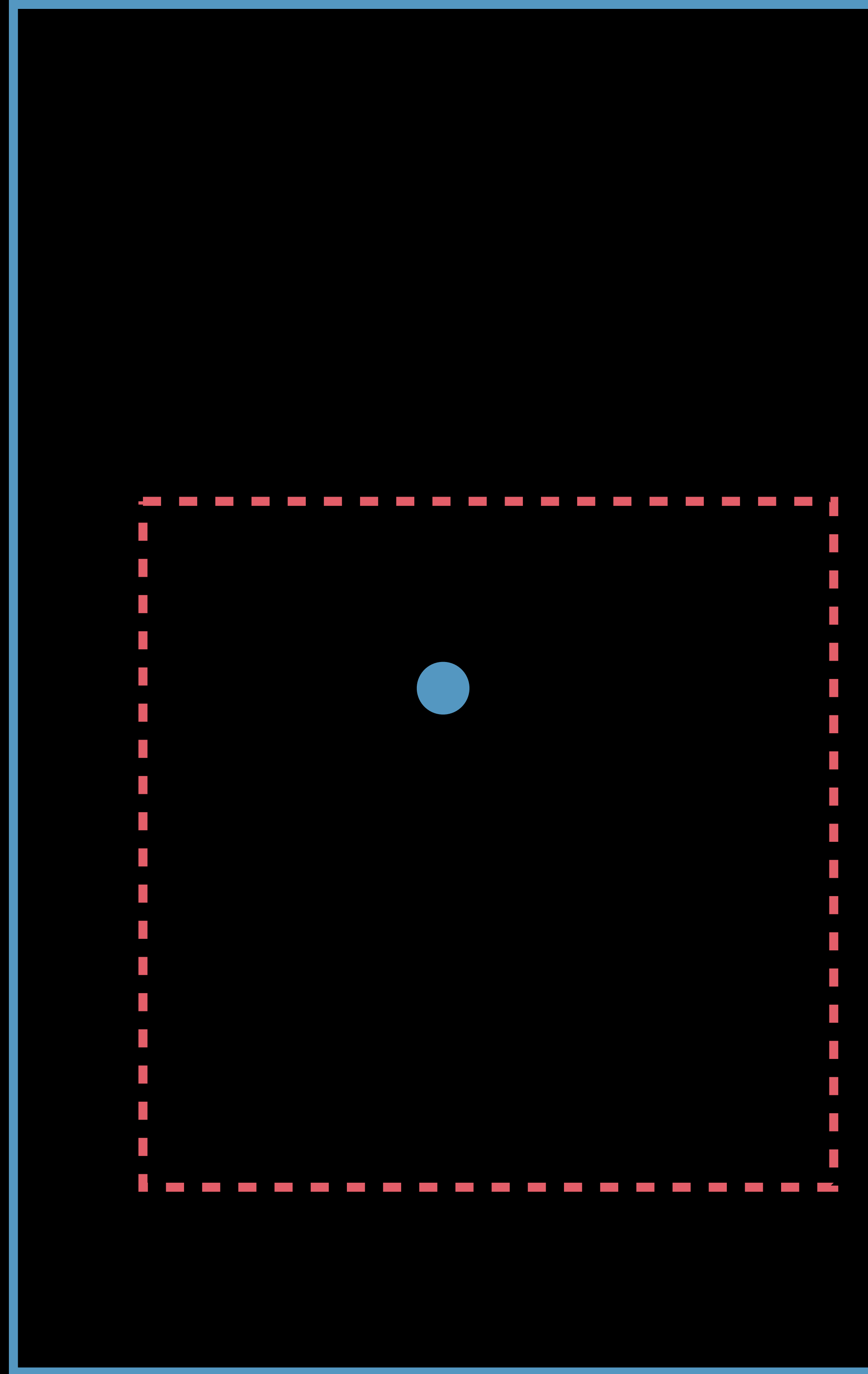

Viewport



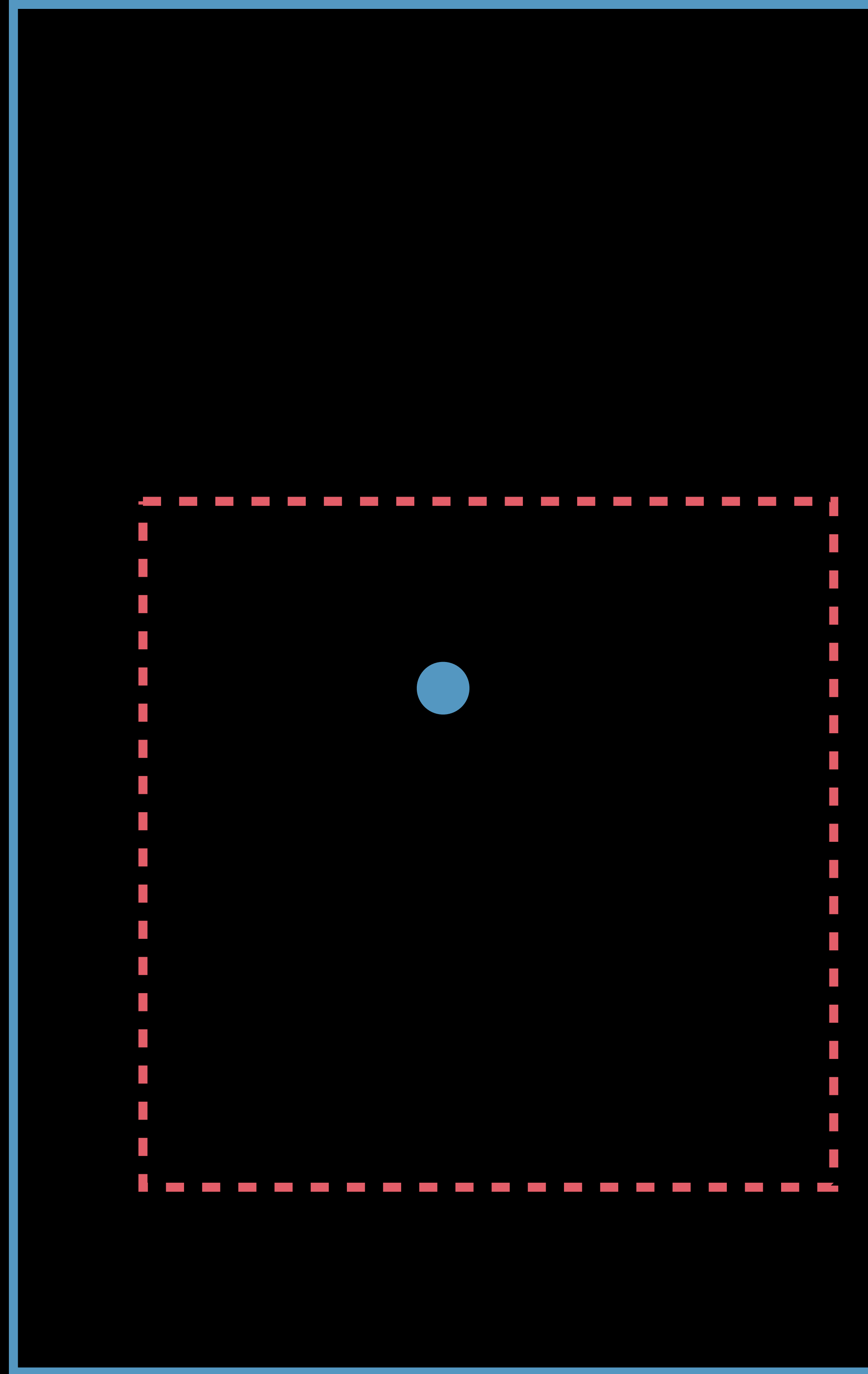
Viewport



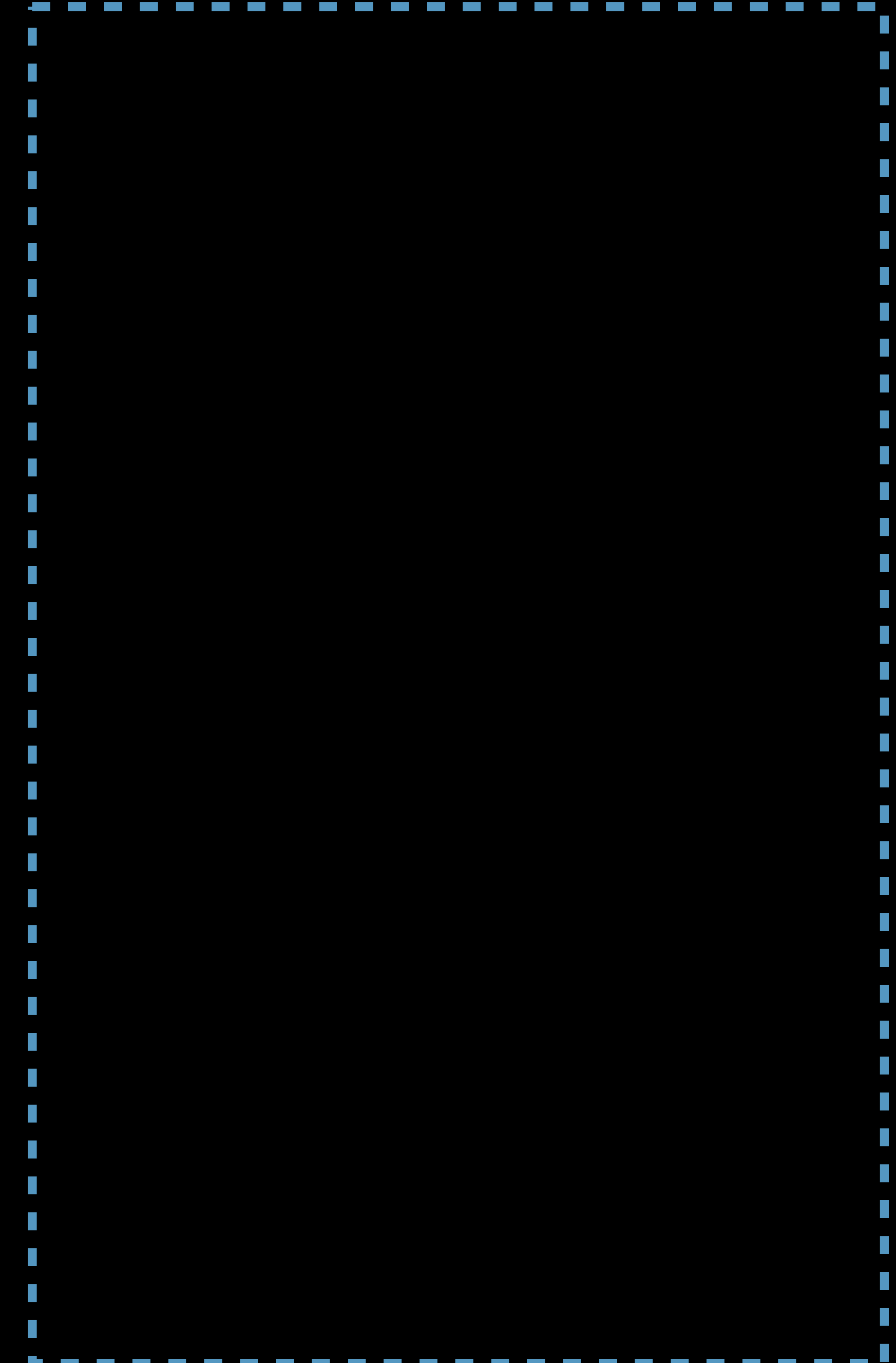
Viewport



Viewport

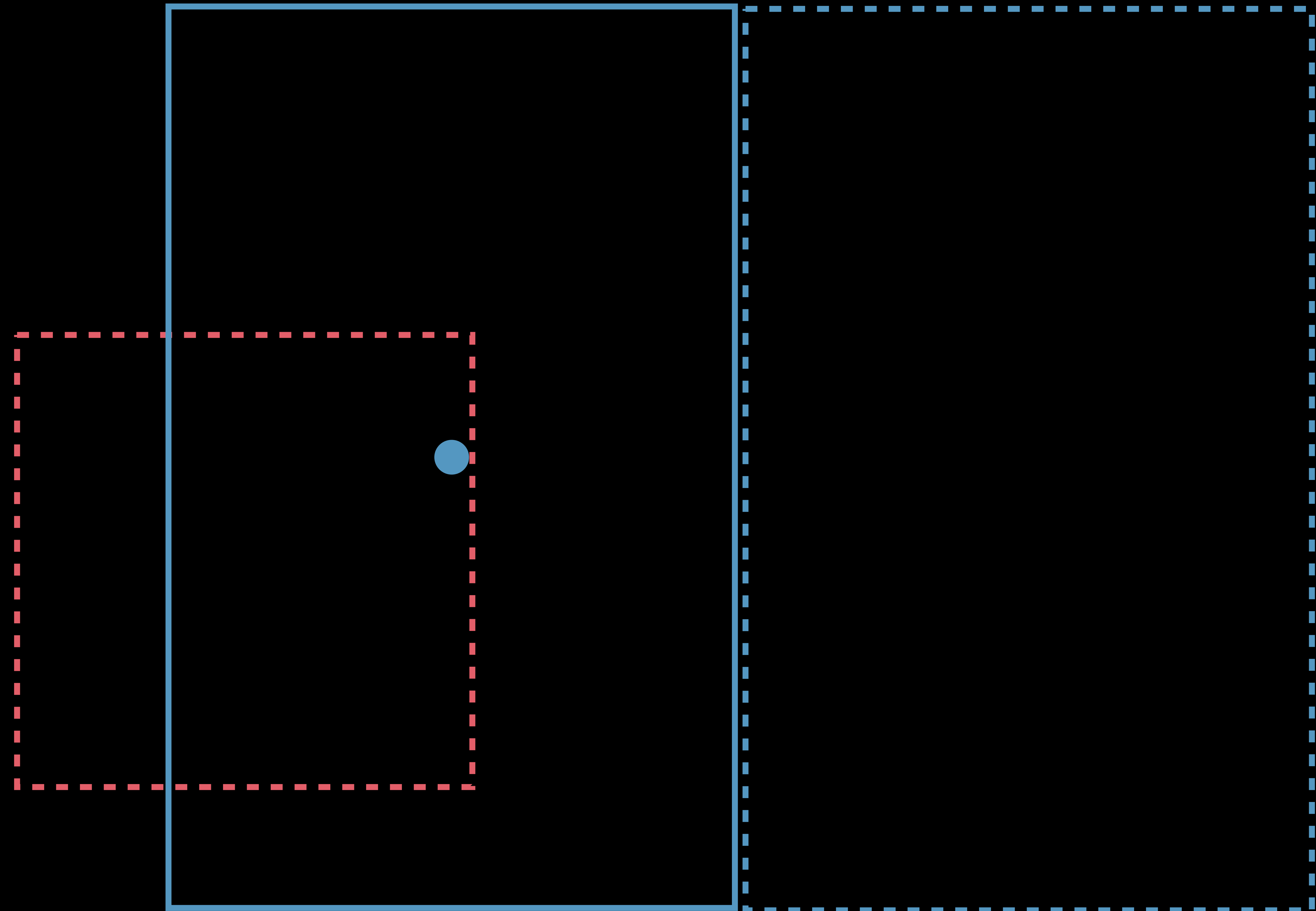


setRegion



Viewport

setRegion



Applying a Camera Boundary in MapKit JS

NEW

```
// Alternative 1: Apply a CoordinateRegion as camera boundary  
map.cameraBoundary = mapkit.CoordinateRegion(coordinate, span);
```

```
// Alternative 2: Apply a MapRect as camera boundary  
map.cameraBoundary = mapkit.MapRect(x, y, width, height);
```


9:41



[Back](#)

Concert in The Park



Apple Maps

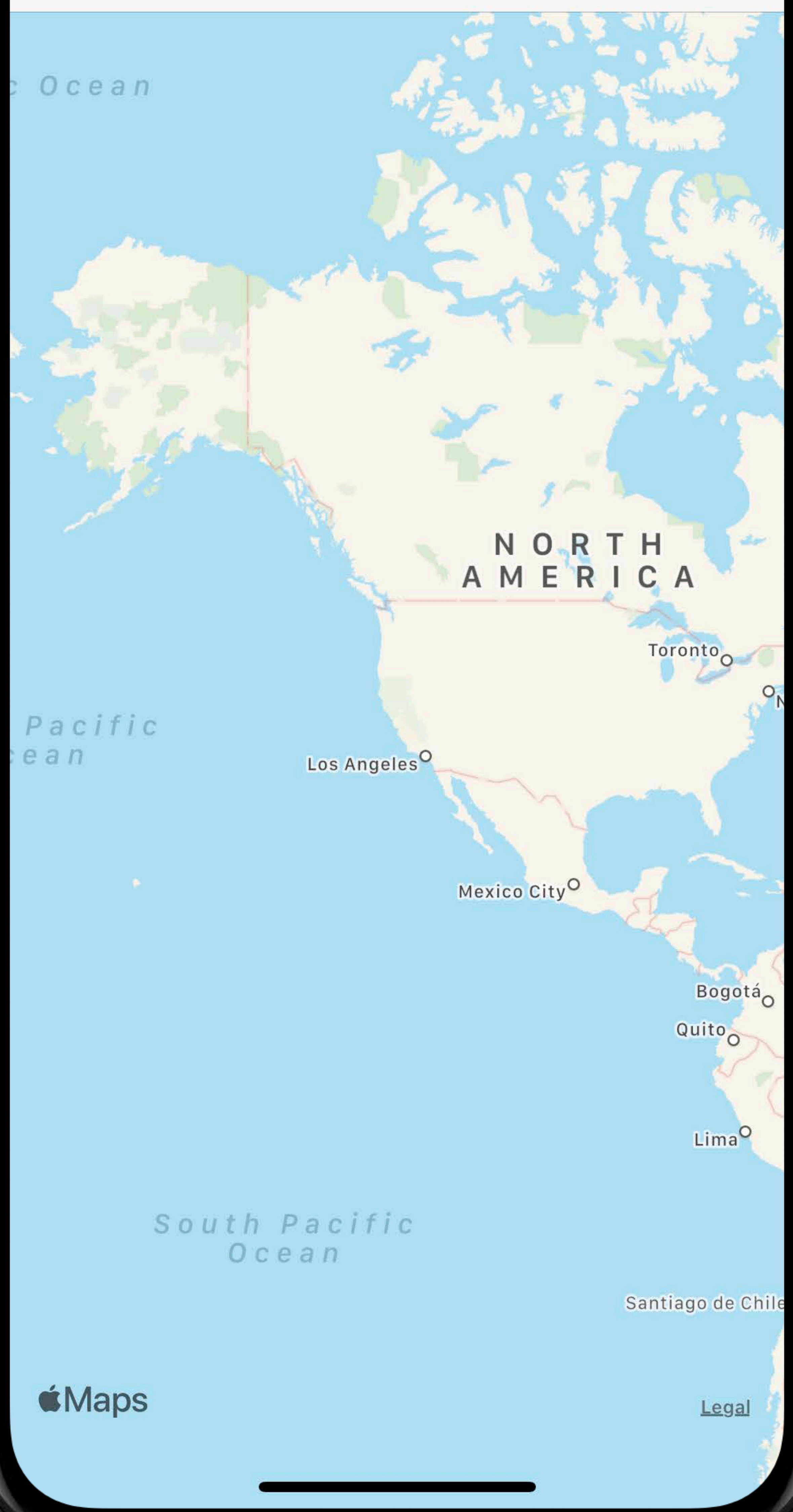
Legal

9:41



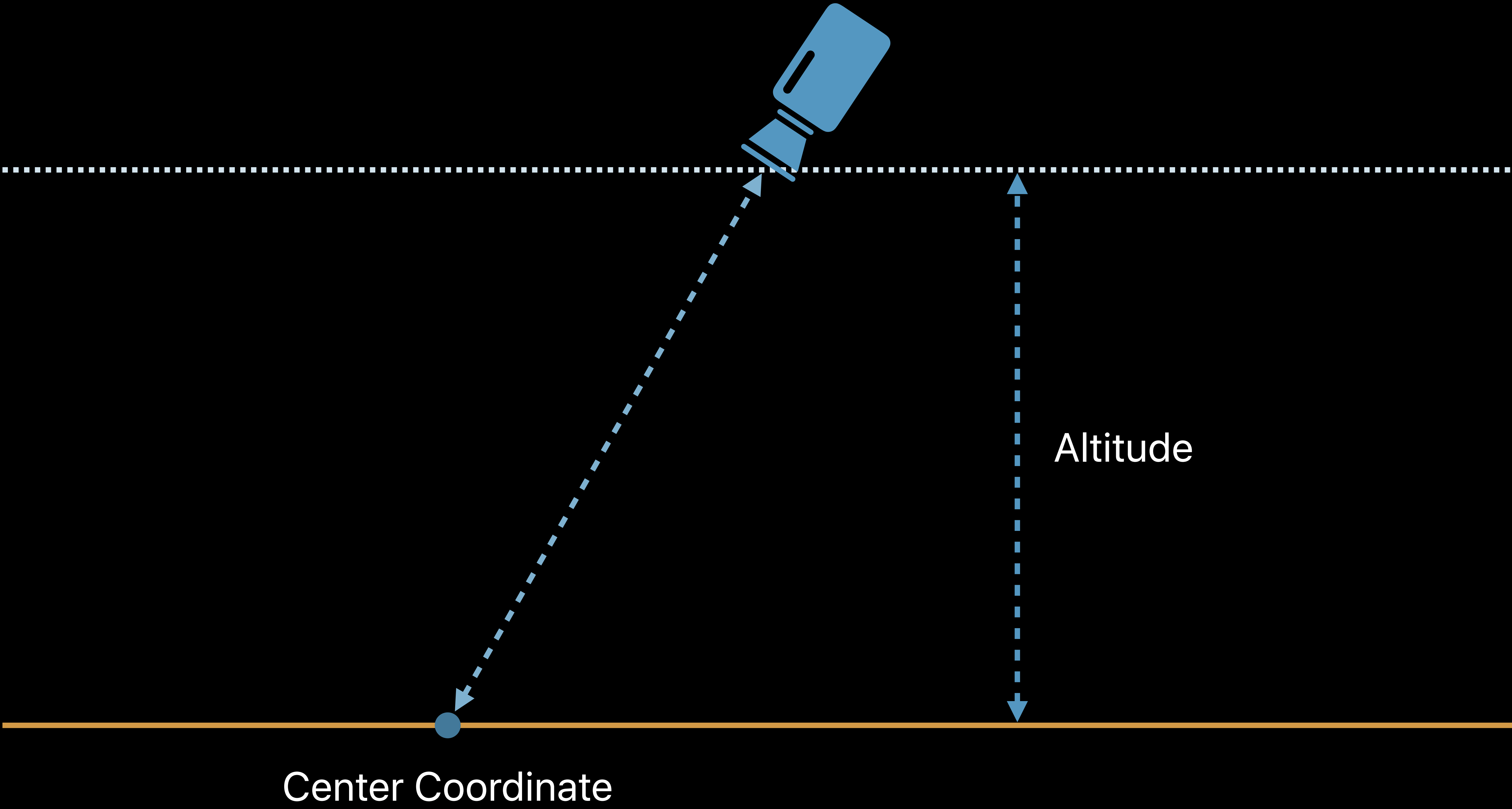
< Back

Concert in The Park

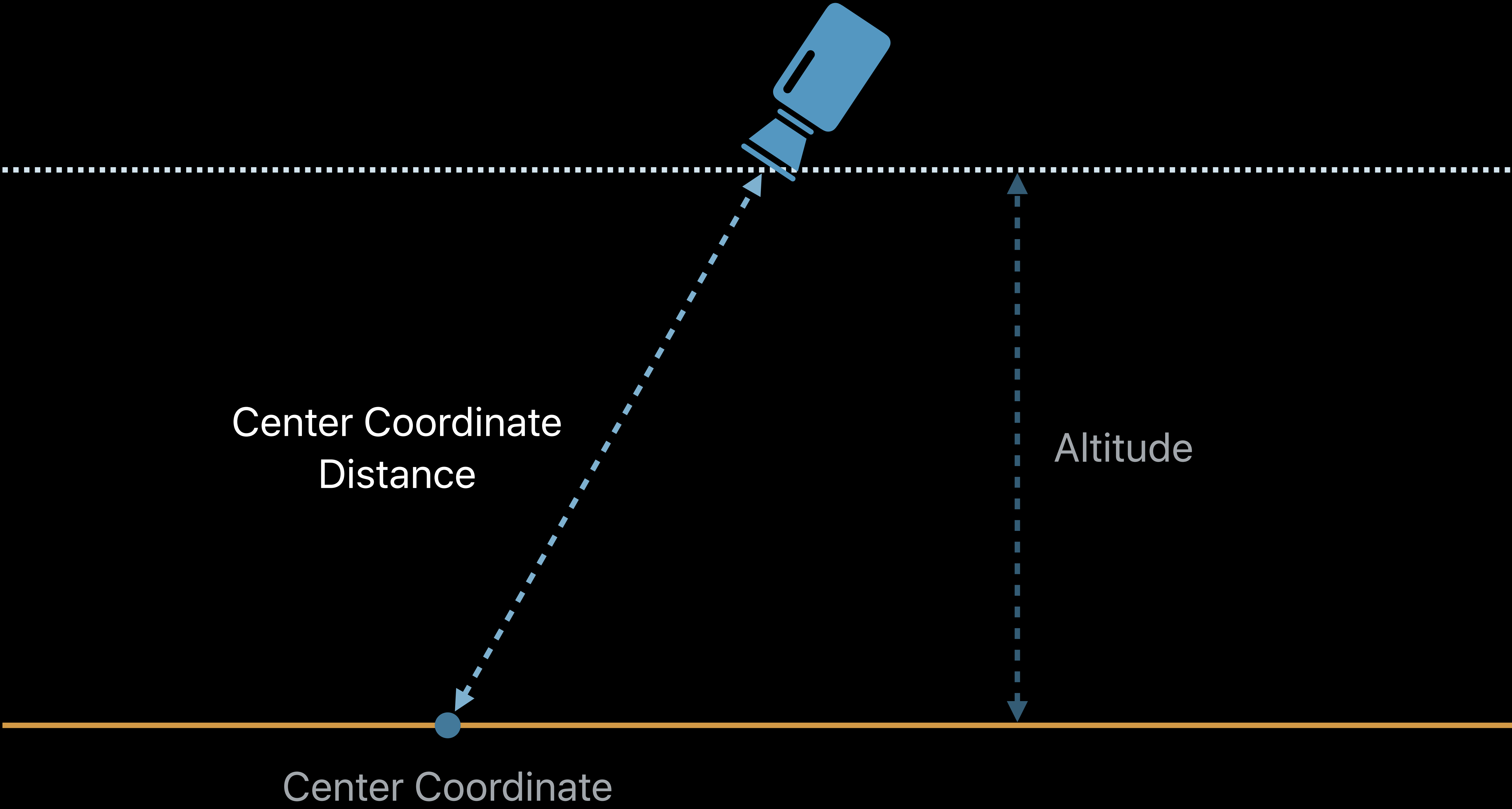


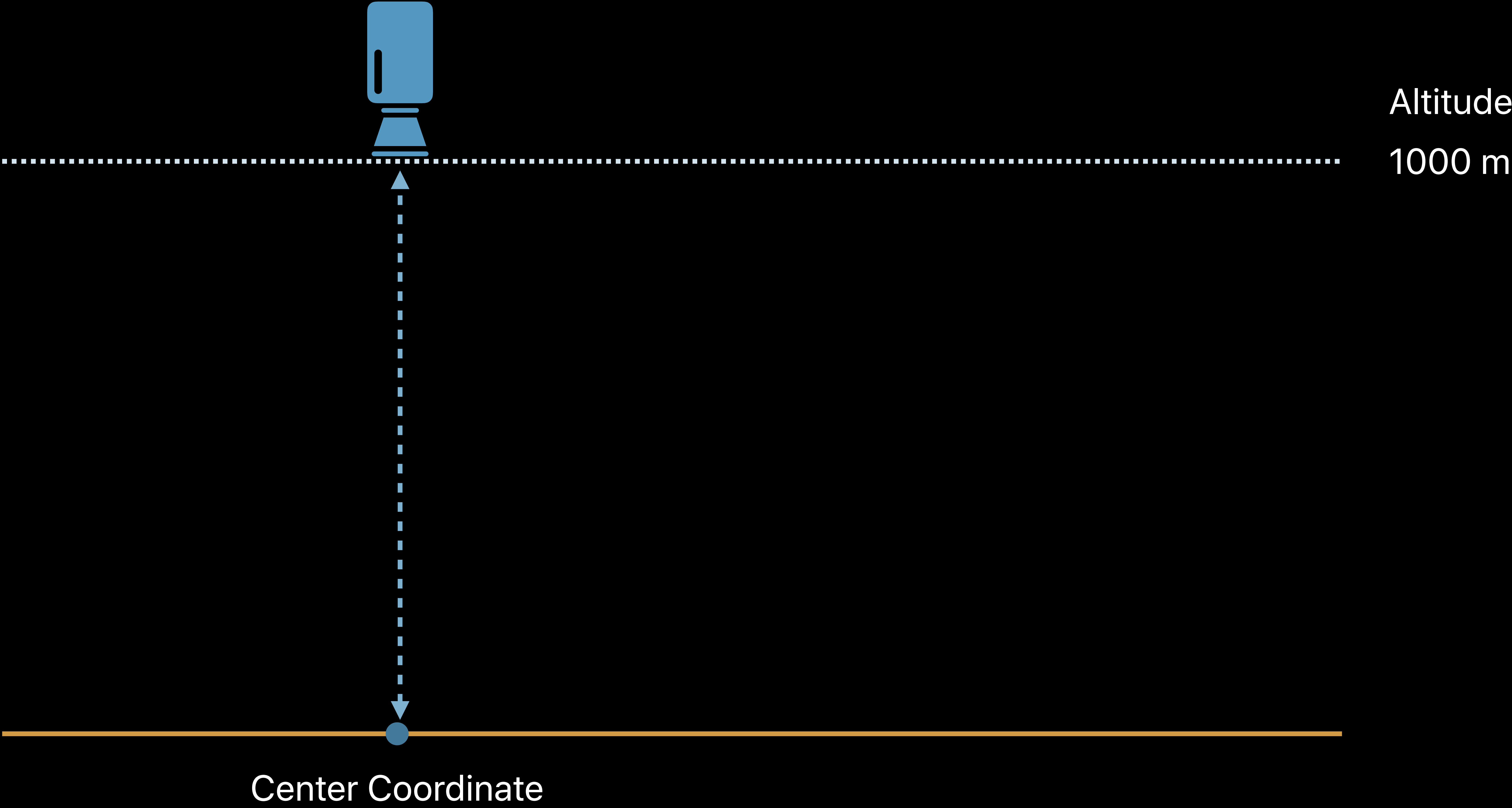
Apple Maps

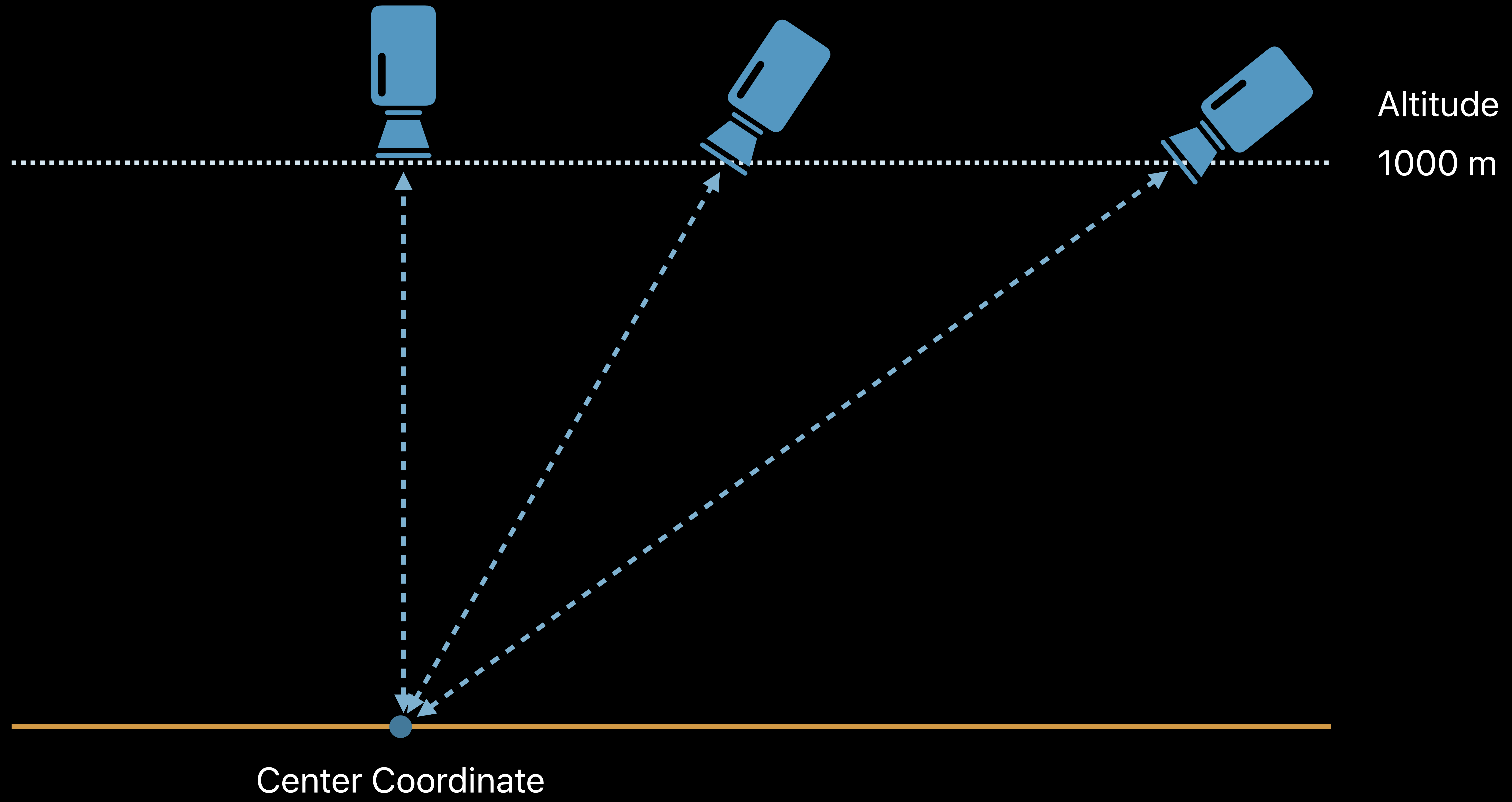
Legal

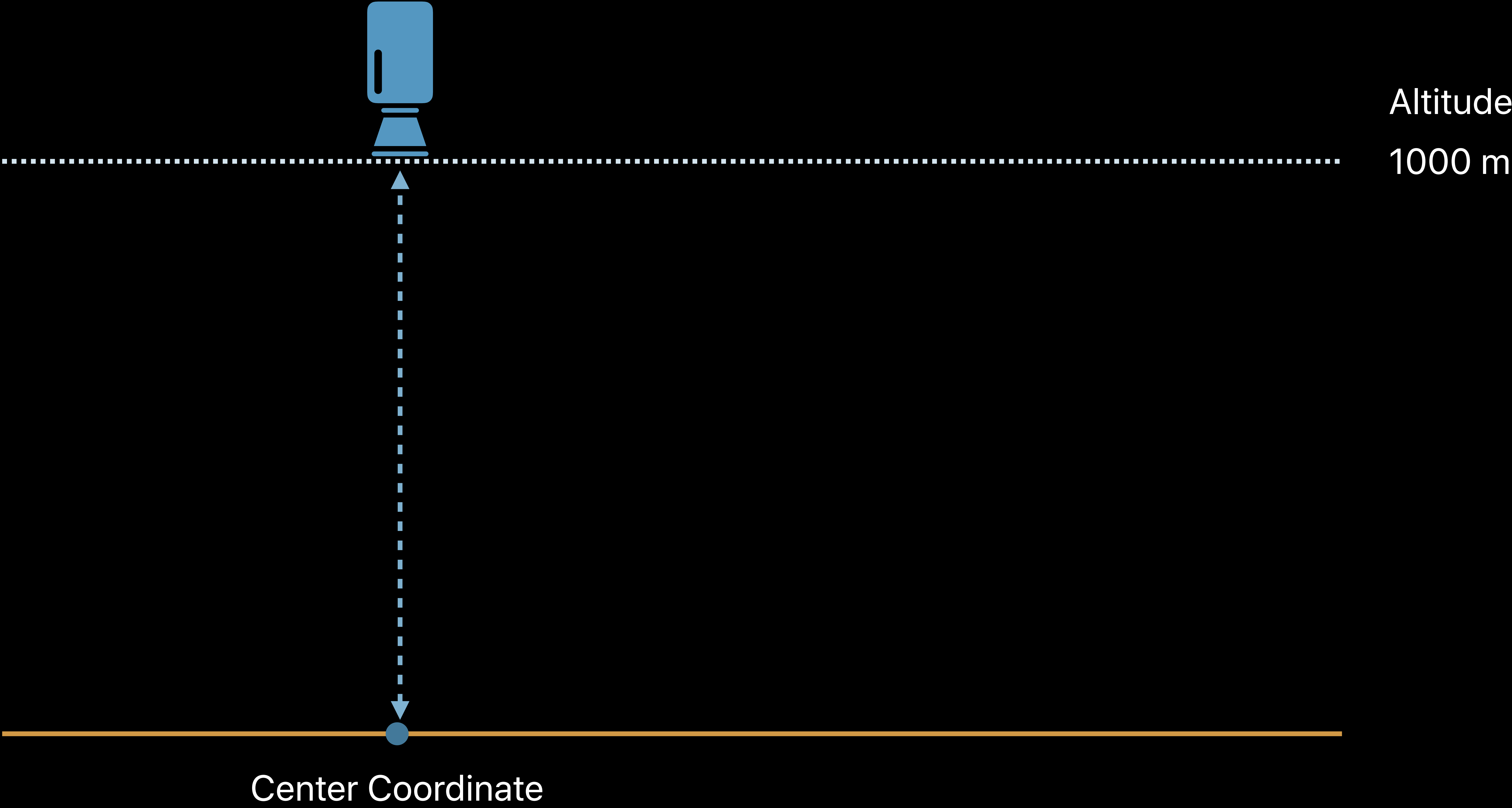


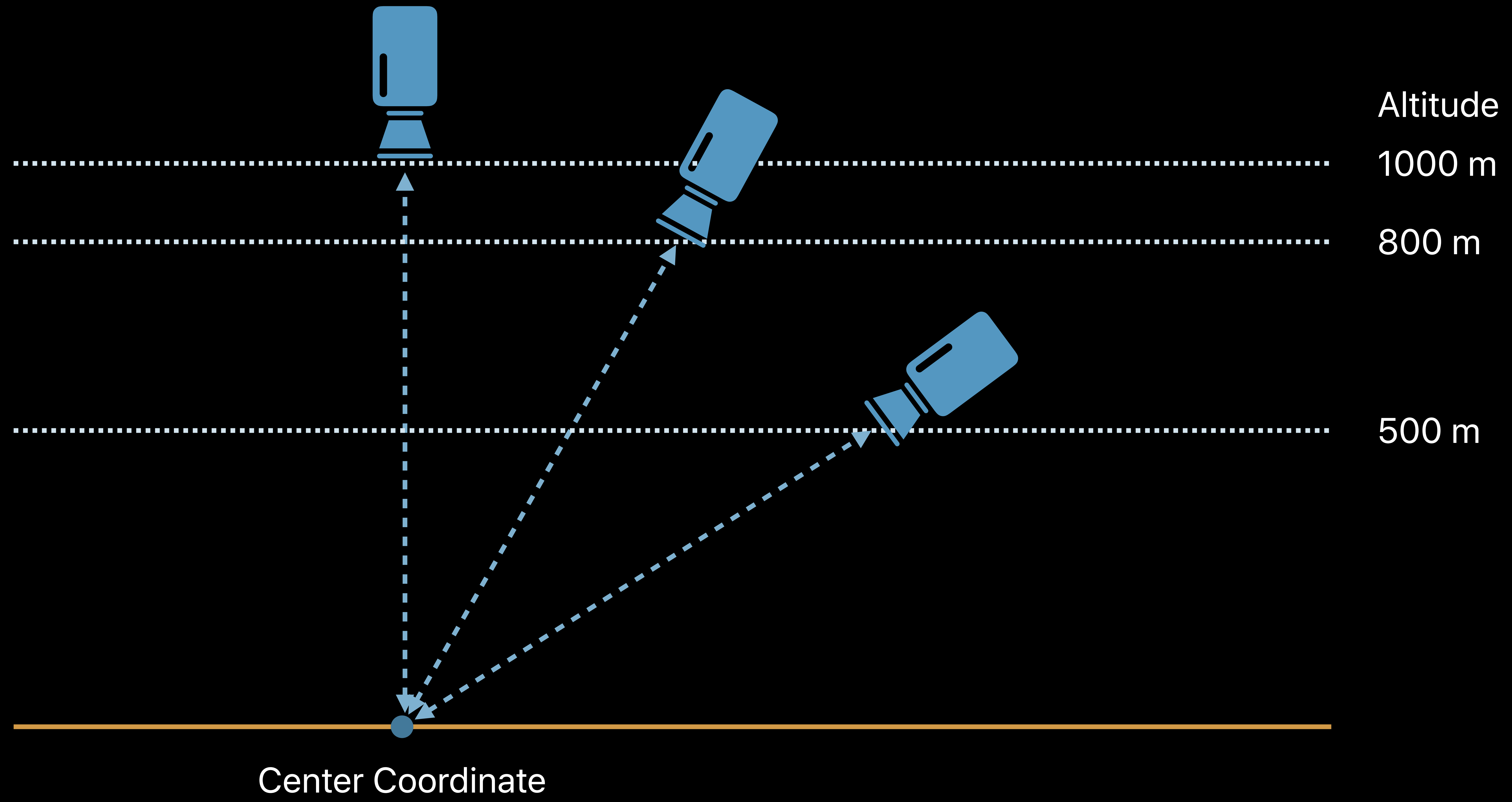
NEW







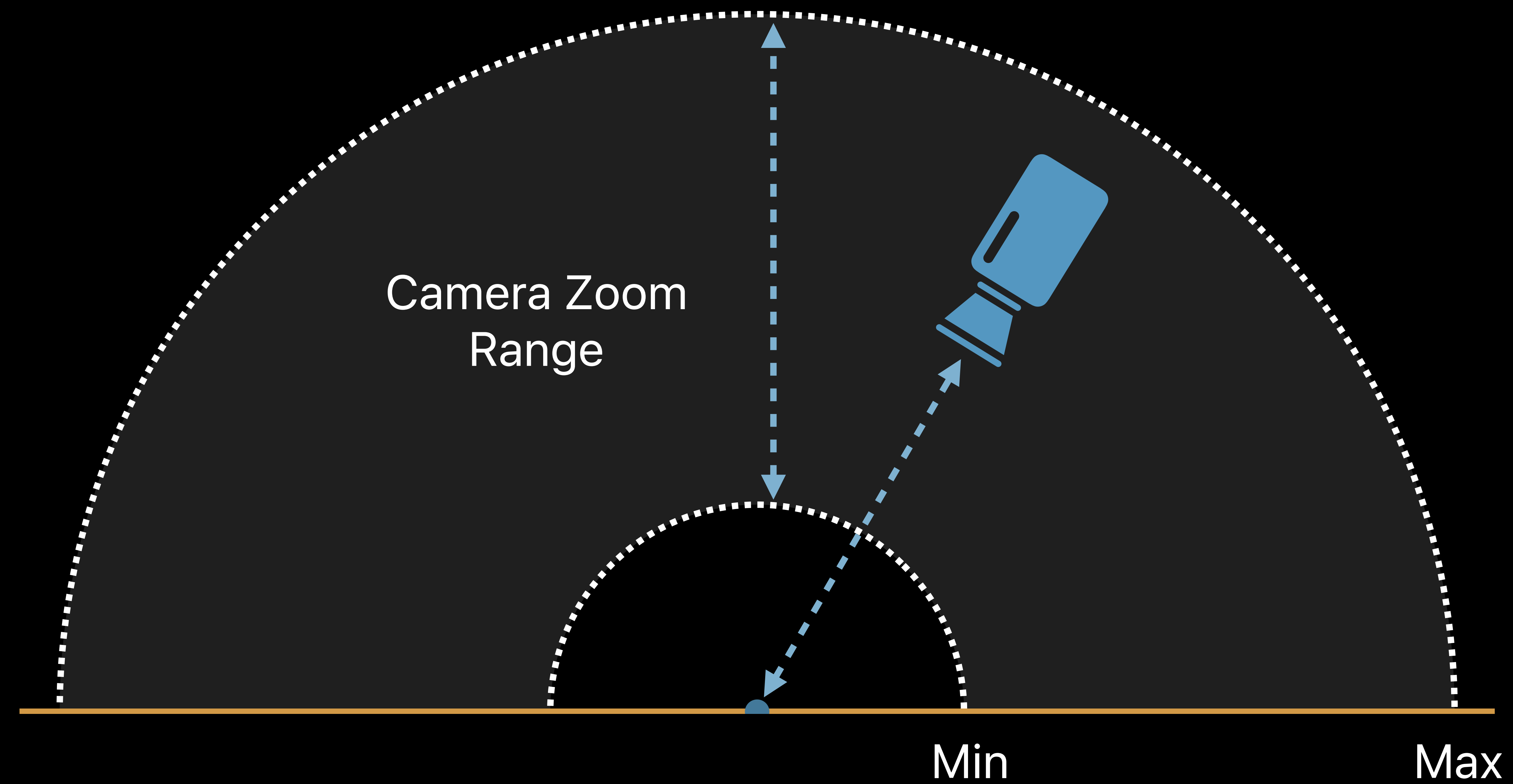




MKMapView.CameraZoomRange

NEW

Defines a minimum and maximum center coordinate distance

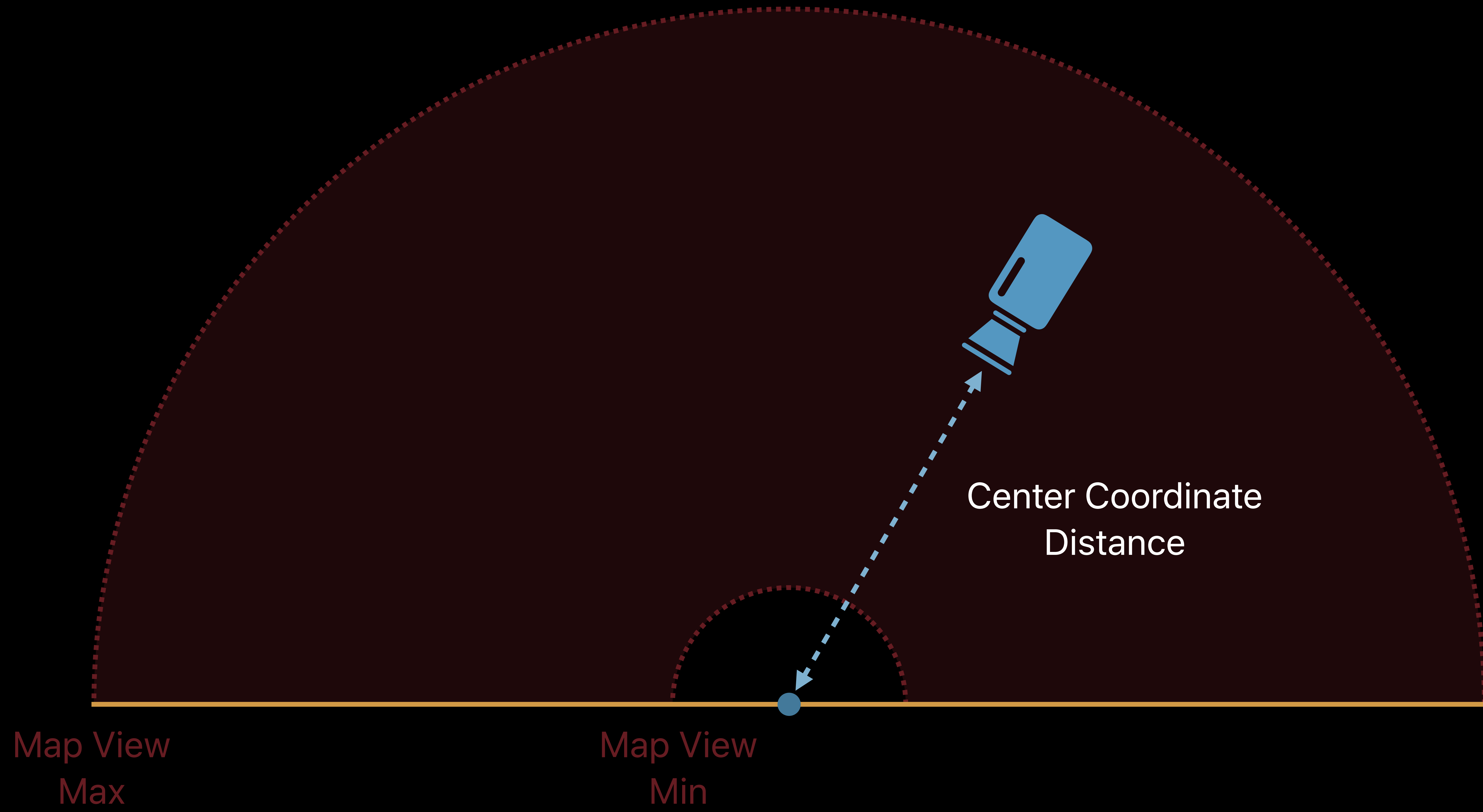


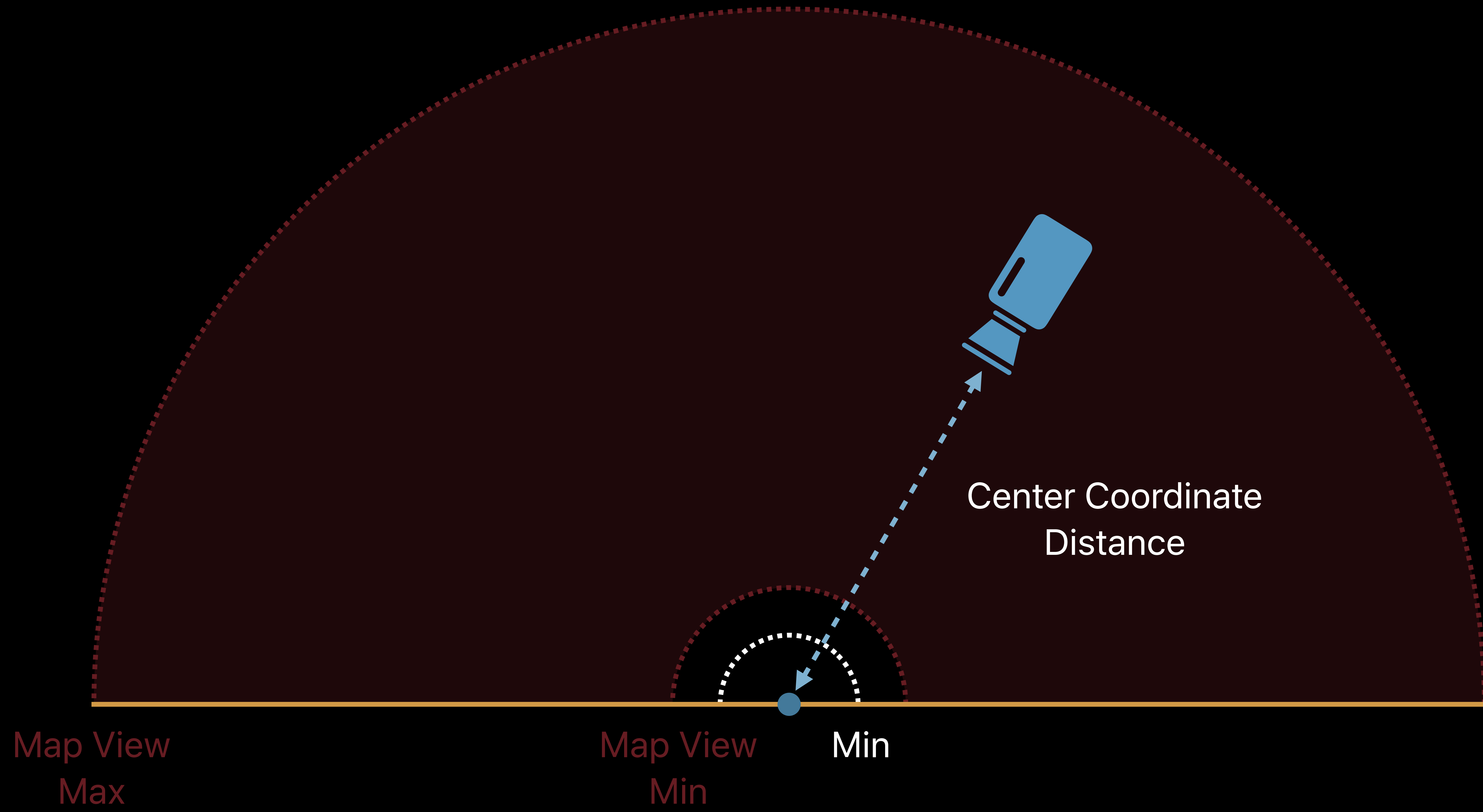
Applying a Camera Zoom Range

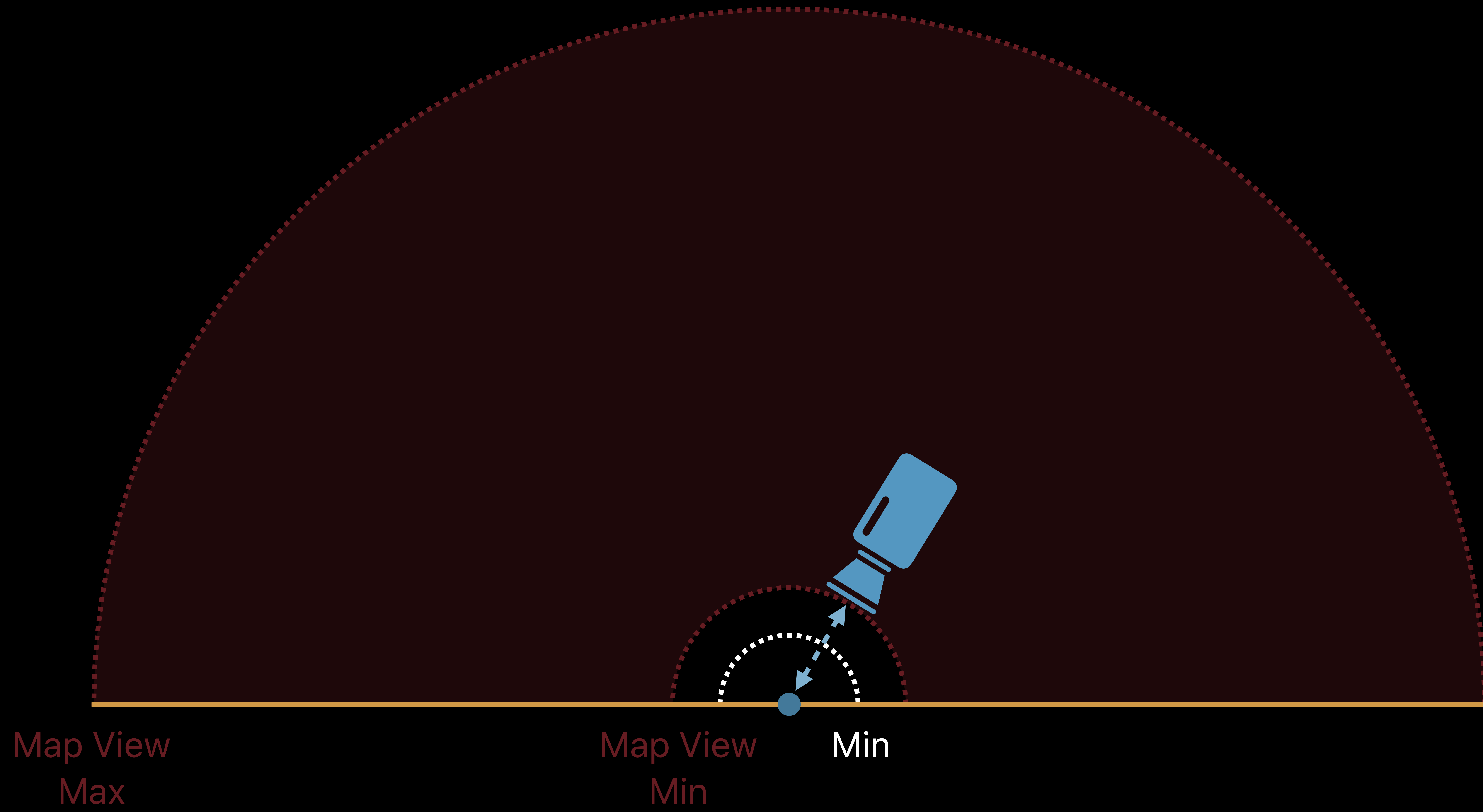
```
// Alternative 1: Create using both min and max center coordinate distances  
mapView.cameraZoomRange = MKMapView.CameraZoomRange(minCenterCoordinateDistance: 500,  
                                                       maxCenterCoordinateDistance: 2000)
```

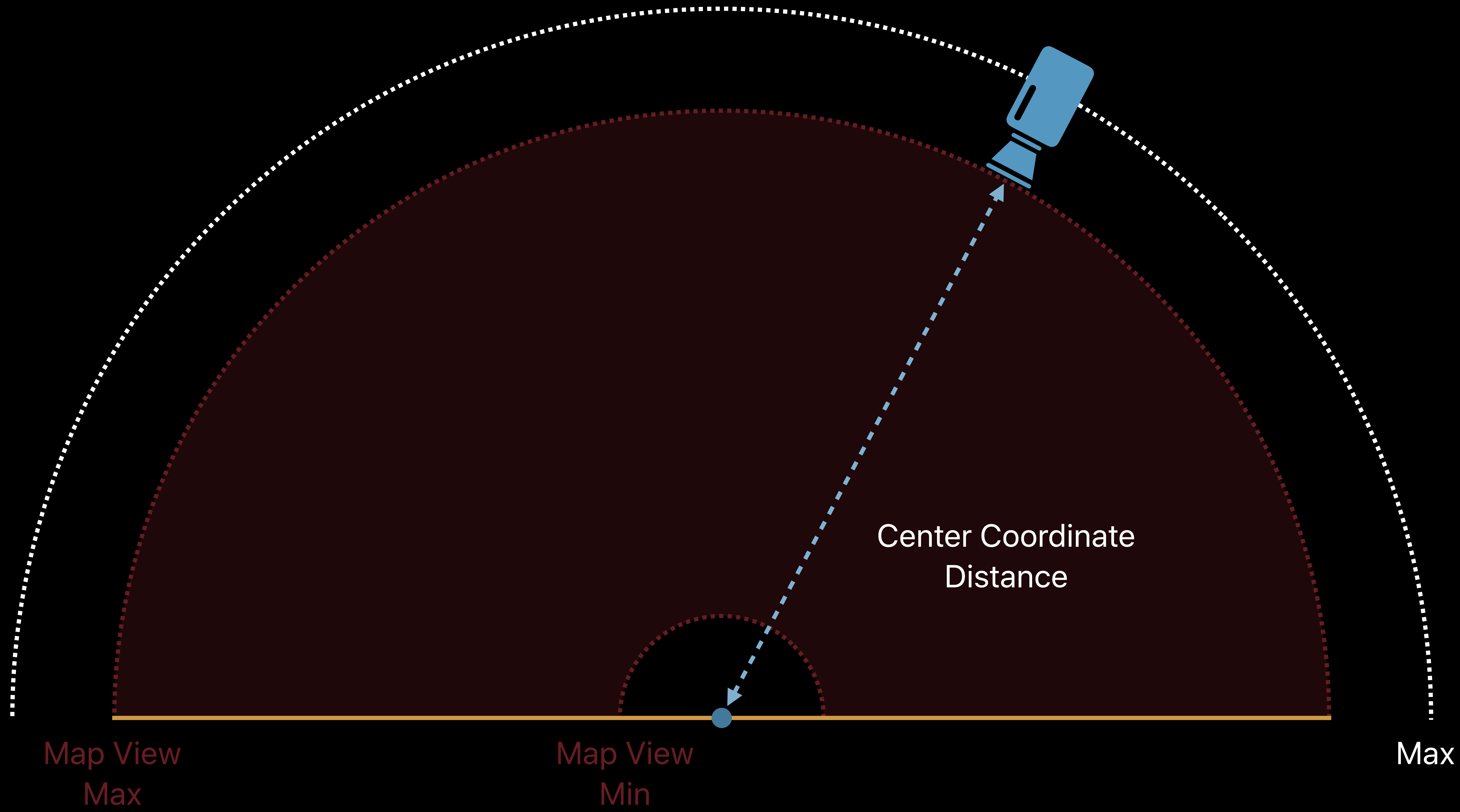
```
// Alternative 2: Create using only a min center coordinate distance  
mapView.cameraZoomRange = MKMapView.CameraZoomRange(minCenterCoordinateDistance: 500)
```

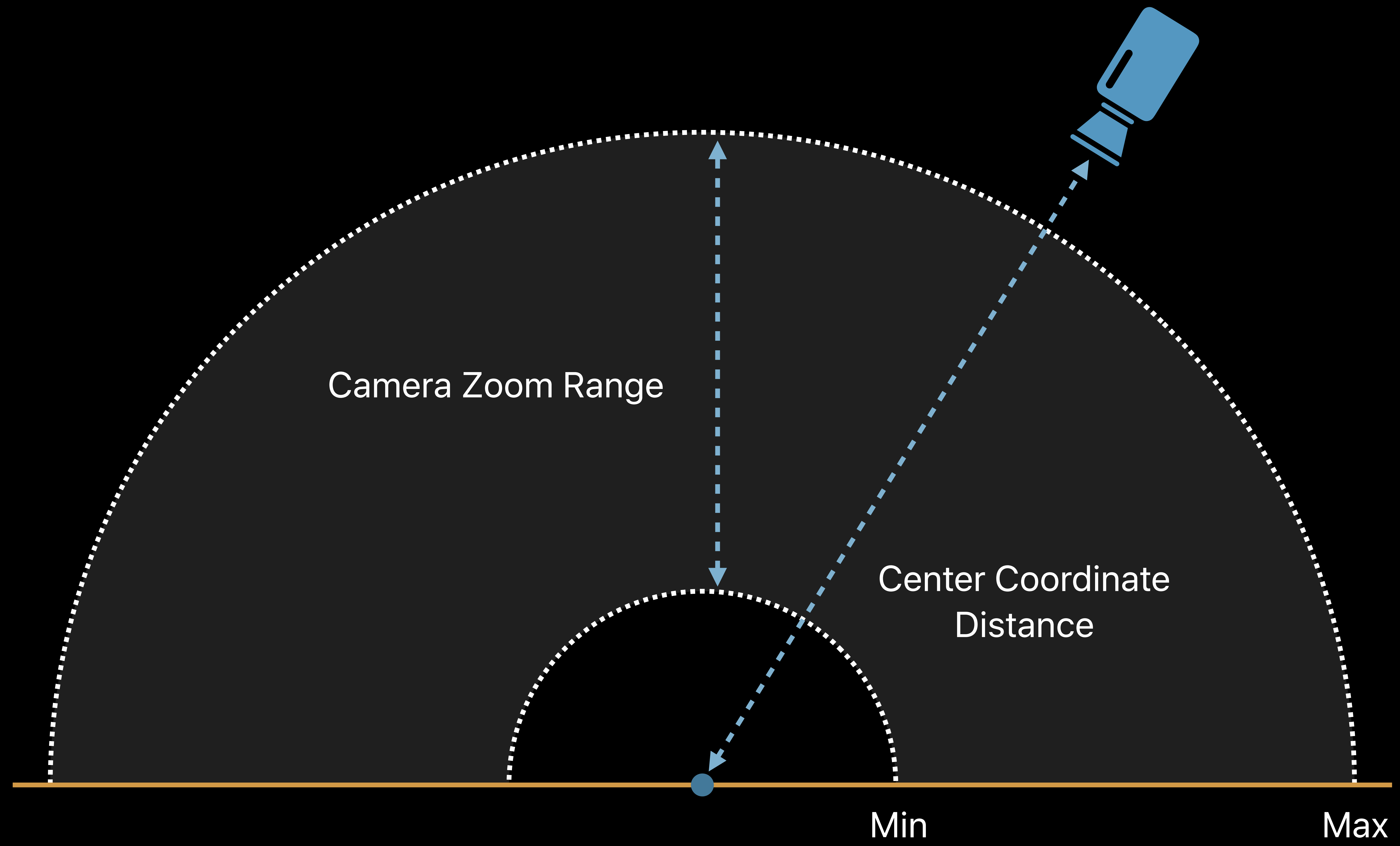
```
// Alternative 3: Create using only a max center coordinate distance  
mapView.cameraZoomRange = MKMapView.CameraZoomRange(maxCenterCoordinateDistance: 500)
```







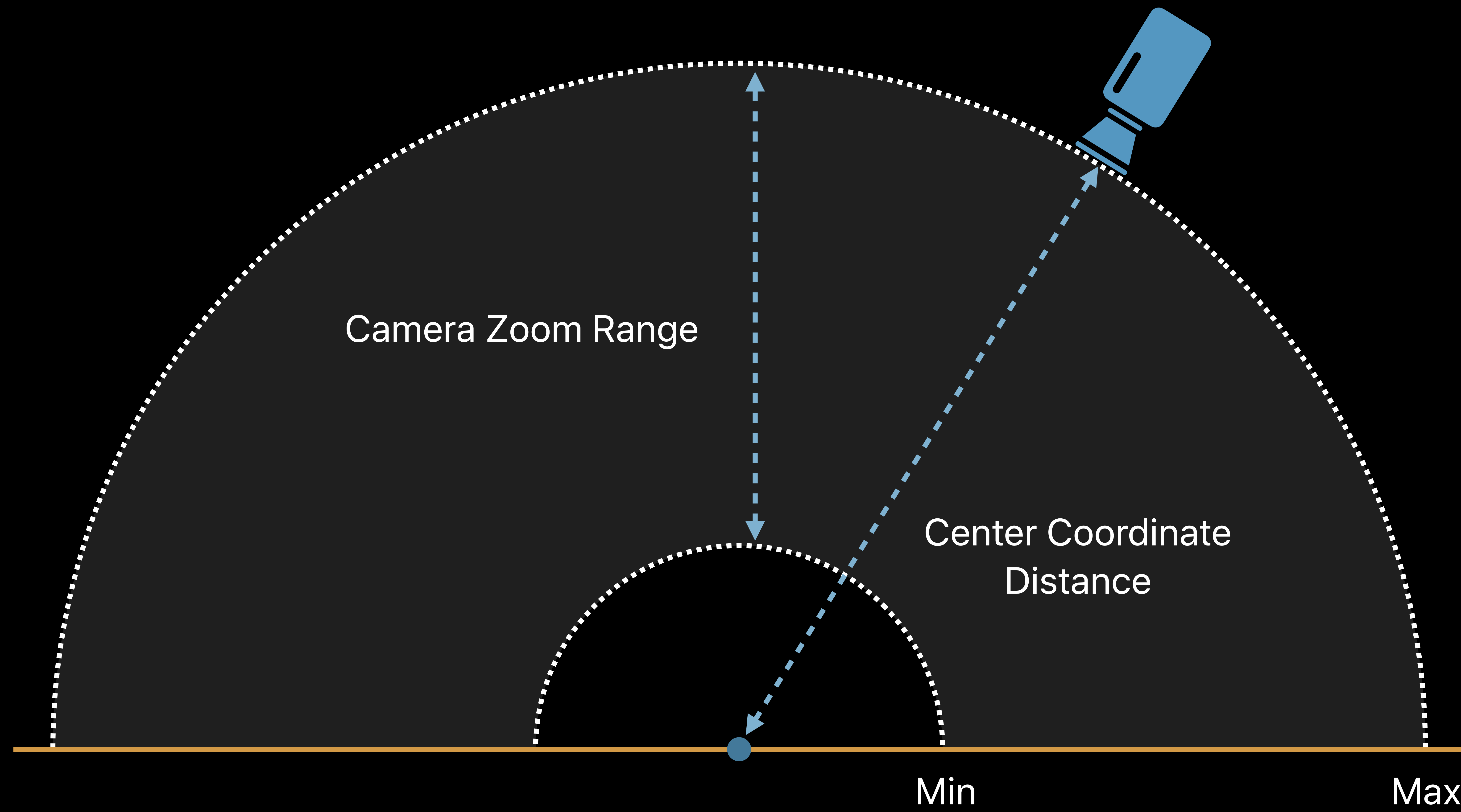


Camera Zoom Range

Center Coordinate Distance

Min

Max



Applying a Camera Zoom Range in MapKit JS

NEW

```
// Alternative 1: Create with both min and max camera distance  
map.cameraZoomRange = new mapkit.CameraZoomRange(500, 2000);
```

```
// Alternative 2: Create with only min camera distance  
map.cameraZoomRange = new mapkit.CameraZoomRange({minCameraDistance: 500});
```

```
// Alternative 3: Create with only max camera distance  
map.cameraZoomRange = new mapkit.CameraZoomRange({maxCameraDistance: 2000});
```


Demo

Applying camera boundaries and zoom ranges

Nalini Shah, MapKit Engineer

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

Agenda

Creating snapshots on the web

Using the map in dark mode

Filtering points of interest in map views

Filtering search and autocompletion results

Improving overlay performance

Decoding and presenting GeoJSON

Taking control of the map view camera

Summary

Summary

Replace maps with snapshots if interaction is not needed

Summary

Replace maps with snapshots if interaction is not needed

Apply point of interest filters to tailor your maps to your needs

Summary

Replace maps with snapshots if interaction is not needed

Apply point of interest filters to tailor your maps to your needs

Filter search and autocomplete results for improved relevance

Summary

Replace maps with snapshots if interaction is not needed

Apply point of interest filters to tailor your maps to your needs

Filter search and autocompletion results for improved relevance

Group your polygons and polylines into multipolygons and multipolylines

Summary

Replace maps with snapshots if interaction is not needed

Apply point of interest filters to tailor your maps to your needs

Filter search and autocompletion results for improved relevance

Group your polygons and polylines into multipolygons and multipolylines

Take advantage of MapKit and MapKit JS's GeoJSON support

Summary

Replace maps with snapshots if interaction is not needed

Apply point of interest filters to tailor your maps to your needs

Filter search and autocompletion results for improved relevance

Group your polygons and polylines into multipolygons and multipolylines

Take advantage of MapKit and MapKit JS's GeoJSON support

Use camera boundaries and zoom ranges to focus on what really matters

More Information

developer.apple.com/wwdc19/236

Adding Indoor Maps to your App and Website

Friday, 2:00

MapKit, MapKit JS, and Indoor Location Lab

Friday, 3:00

Introducing the Indoor Map Program

WWDC 2019

 WWDC19