What’s New in MapKit

Session 206

Sumit Lonkar Software Engineer Maps Team
Elisabeth Lindkvist Software Engineer Maps Team
Agenda

What’s New in MapKit
Agenda

What’s New in MapKit

Improvements to Existing API
Agenda

What’s New in MapKit

Improvements to Existing API

Transit
Agenda

What’s New in MapKit

Improvements to Existing API
Transit
Flyover
Agenda

What’s New in MapKit

Improvements to Existing API
Transit
Flyover
Demos
Improvements to Existing API
Pin Color Customizations

Improvements to existing API
Pin Color Customizations

Improvements to existing API
Pin Color Customizations

Improvements to existing API
Pin Color Customizations

Improvements to existing API
Pin Color Customizations

Improvements to existing API

New API to customize pin colors
Pin Color Customizations

Improvements to existing API

New API to customize pin colors

- On iOS — `@availability(iOS 9.0, *) var pinTintColor: UIColor!`
Pin Color Customizations

Improvements to existing API

New API to customize pin colors

- On iOS — `@availability(iOS 9.0, *) var pinTintColor: UIColor!`
- On OSX — `@availability(OSX 10.11, *) var pinTintColor: NSColor!`
Pin Color Customizations

Improvements to existing API

```swift
var pinColor: MKPinAnnotationColor
```
Callout Customizations

Improvements to existing API
Callout Customizations

Improvements to existing API

Callouts
Callout Customizations

Improvements to existing API

Callouts
Callout Customizations

Improvements to existing API

Callouts

• Title
Callout Customizations

Improvements to existing API

Callouts

• Title
• Subtitle
Callout Customizations

Improvements to existing API

Callouts

• Title
• Subtitle
• Right accessory view
Callout Customizations

Improvements to existing API

Callouts
• Title
• Subtitle
• Right accessory view
• Left accessory view
Callout Customizations

Improvements to existing API

Callouts
- Title
- Subtitle
- Right accessory view
- Left accessory view

![Map with callout example](image)
Callout Customizations
Improvements to existing API

Callouts
Content

Moscone West
Welcome to session 206. We are really excited to show new co...
Callout Customizations

Improvements to existing API

Callouts
Content
Customize to fit your apps need
Callout Customizations
Improvements to existing API
Callout Customizations

Improvements to existing API
Callout Customizations

Improvements to existing API
Callout Customizations

Improvements to existing API
Callout Customizations

Improvements to existing API

New API to customize callouts
Callout Customizations
Improvements to existing API

New API to customize callouts

- On iOS

```swift
@availability(iOS 9.0, *) var detailCalloutAccessoryView: UIView?
```
Callout Customizations

Improvements to existing API

New API to customize callouts

- On iOS

  `@availability(iOS 9.0, *) var detailCalloutAccessoryView: UIView?`

- On OSX

  `@availability(OSX 10.11, *) var detailCalloutAccessoryView: NSView?`
Callout Customizations

Improvements to existing API

Sample Code
Callout Customizations

Improvements to existing API

Sample Code

```swift
func mapView(mapView: MKMapView, viewForAnnotation annotation: MKAnnotation) -> MKAnnotationView? {
    ...
    view = MKPinAnnotationView(annotation: annotation, reuseIdentifier: identifier)
    view.canShowCallout = true
    view.detailCalloutAccessoryView = UIImageView(image: UIImage(named: "tajMahal"))
    ...
}
```
Callout Customizations

Improvements to existing API

Sample Code

```swift
func mapView(mapView: MKMapView, viewForAnnotation annotation: MKAnnotation) -> MKAnnotationView? {
    ...
    view = MKPinAnnotationView(annotation: annotation, reuseIdentifier: identifier)
    view.canShowCallout = true
    view.detailCalloutAccessoryView = UIImageView(image: UIImage(named: "tajMahal"))
    ...
}
```
Callout Customizations

Improvements to existing API

Sample Code

```swift
func mapView(mapView: MKMapView, viewForAnnotation annotation: MKAnnotation) -> MKAnnotationView? {
    ...
    view = MKPinAnnotationView(annotation: annotation, reuseIdentifier: identifier)
    view.canShowCallout = true
    view.detailCalloutAccessoryView = UIImageView(image: UIImage(named: "tajMahal"))
    ...
}
```
Map Customizations

Improvements to existing API
Map Customizations

Improvements to existing API

New API to customize your Map View
Map Customizations
Improvements to existing API

New API to customize your Map View
  • Show / Hide Traffic
    `mapView.showsTraffic = true`
Map Customizations

Improvements to existing API

New API to customize your Map View

• Show / Hide Traffic
  ```javascript
  mapView.showsTraffic = true
  ```

• Show / Hide Scale
  ```javascript
  mapView.showsScale = true
  ```
Map Customizations

Improvements to existing API

New API to customize your Map View

- Show / Hide Traffic
  ```javascript
  mapView.showsTraffic = true
  ```
- Show / Hide Scale
  ```javascript
  mapView.showsScale = true
  ```
- Show / Hide Compass
  ```javascript
  mapView.showsCompass = true
  ```
Map Customizations

Improvements to existing API

New API to customize your Map View

- Show / Hide Traffic
  ```swift
  mapView.showsTraffic = true
  ```
- Show / Hide Scale
  ```swift
  mapView.showsScale = true
  ```
- Show / Hide Compass
  ```swift
  mapView.showsCompass = true
  ```
- Similar to existing API
  ```swift
  var showsPointsOfInterest: Bool
  var showsBuildings: Bool
  ```
Miscellaneous

Improvements to MapKit
Miscellaneous

Improvements to MapKit

Time zone support
Miscellaneous

Improvements to MapKit

Time zone support

CLGeocoder
Miscellaneous

Improvements to MapKit

Time zone support

 CLGeocoder
 MKLocalSearch
Miscellaneous

Improvements to MapKit

Time zone support

CLGeocoder
MKLocalSearch

Swift improvements
Miscellaneous

Improvements to MapKit

Time zone support
- CLGeocoder
- MKLocalSearch

Swift improvements

WatchKit support
Transit
What Transit Means to MapKit

Transit
What Transit Means to MapKit

Maps Transit features

Transit in Maps App
What Transit Means to MapKit
Maps Transit features

Transit in Maps App
User experience is critical
What Transit Means to MapKit

Maps Transit features

Transit in Maps App
User experience is critical
Leverage Maps to provide transit direction in your app
Transit ETA Requests

Transit
Transit ETA Requests

Transit

MKDirectionsTransportType

... static var Automobile: MKDirectionsTransportType { get } static var Walking: MKDirectionsTransportType { get } static var Any: MKDirectionsTransportType { get }

...
Transit ETA Requests

Transit

MKDirectionsTransportType

... static var Automobile: MKDirectionsTransportType { get } static var Walking: MKDirectionsTransportType { get } static var Transit: MKDirectionsTransportType { get } static var Any: MKDirectionsTransportType { get }

...
Transit ETA Requests

Transit

MKDirectionsTransportType

... static var Automobile: MKDirectionsTransportType { get } static var Walking: MKDirectionsTransportType { get } static var Transit: MKDirectionsTransportType { get } static var Any: MKDirectionsTransportType { get } ...

Note — New type is only available for ETA Requests
Transit ETA Requests

Transit

MKDirectionsTransportType

In addition to ETA you will get
Transit ETA Requests

Transit

MKDirectionsTransportType

In addition to ETA you will get

• Expected Arrival Time
Transit ETA Requests

Transit

MKDirectionsTransportType

In addition to ETA you will get

• Expected Arrival Time
• Expected Departure Time
Transit ETA Requests

Transit

Sample Code
func getTransitETA(){
    let request = MKDirectionsRequest()
    /*            Set Source            */
    /*            Set Destination        */
    //Set Transport Type to be Transit
    request.transportType = MKDirectionsTransportType.Transit
    let directions = MKDirections(request: request)
    directions.calculateETAWithCompletionHandler { response, error in
        // Handle Response
    }
}
Transit ETA Requests

Transit

Sample Code

```swift
func getTransitETA()
{
    let request = MKDirectionsRequest()
    /* Set Source */
    /* Set Destination */

    // Set Transport Type to be Transit
    request.transportType = MKDirectionsTransportType.Transit
    let directions = MKDirections(request: request)
    directions.calculateETAWithCompletionHandler { response, error in
        // Handle Response
    }
}
```
Transit ETA Requests

Transit

Sample Code

```swift
func getTransitETA()
{
    let request = MKDirectionsRequest()
    /*         Set Source            */
    /*         Set Destination        */
    
    //Set Transport Type to be Transit
    request.transportType = MKDirectionsTransportType.Transit
    let directions = MKDirections(request: request)
    directions.calculateETAWithCompletionHandler { response, error in
        // Handle Response
    }
}
```
func getTransitETA() {
    let request = MKDirectionsRequest()
    /* Set Source */
    /* Set Destination */

    // Set Transport Type to be Transit
    request.transportType = MKDirectionsTransportType.Transit
    let directions = MKDirections(request: request)
    directions.calculateETAWithCompletionHandler { response, error in
        // Handle Response
    }
}
Sample Code

```swift
func getTransitETA()
{
    let request = MKDirectionsRequest()
    /*         Set Source            */
    /*         Set Destination        */
    /*
    
    //Set Transport Type to be Transit
    request.transportType = MKDirectionsTransportType.Transit
    let directions = MKDirections(request: request)
    directions.calculateETAWithCompletionHandler { response, error in
    // Handle Response
    }
}
```
func getTransitETA()
{
    let request = MKDirectionsRequest()
    /*          Set Source            */
    /*          Set Destination        */
    request.transportType = MKDirectionsTransportType.Transit
    let directions = MKDirections(request: request)
    directions.calculateETAWithCompletionHandler { response, error in
        // Handle Response
    }
}
Launch Maps into Transit Mode

Transit
Launch Maps into Transit Mode

Transit

MKLaunchOptionsDirectionsModeKey
Launch Maps into Transit Mode

Transit

MKLaunchOptionsDirectionsModeKey

@availability(iOS 6.0, *)
let MKLaunchOptionsDirectionsModeDriving: String

@availability(iOS 6.0, *)
let MKLaunchOptionsDirectionsModeWalking: String
Launch Maps into Transit Mode

Transit

MKLaunchOptionsDirectionsModeKey

@availability(iOS 6.0, *)
let MKLaunchOptionsDirectionsModeDriving: String

@availability(iOS 6.0, *)
let MKLaunchOptionsDirectionsModeWalking: String

@availability(iOS 9.0, *)
let MKLaunchOptionsDirectionsModeTransit: String
Launch Maps into Transit Mode

Transit

MKLaunchOptionsDirectionsMode

Sample code
Launch Maps into Transit Mode

Transit

MKLaunchOptionsDirectionsMode

Sample code

```swift
func openInMapsTransit(coord: CLLocationCoordinate2D) {
    var placemark = MKPlacemark(coordinate: coord, addressDictionary: nil)
    var mapItem = MKMapItem(placemark: placemark)
    let launchOptions = [MKLaunchOptionsDirectionsModeKey: MKLaunchOptionsDirectionsModeTransit]
    mapItem.openInMapsWithLaunchOptions(launchOptions)
}
```
Launch Maps into Transit Mode

Transit

MKLaunchOptionsDirectionsMode

Sample code

```swift
func openInMapsTransit(coord: CLLocationCoordinate2D) {
    var placemark = MKPlacemark(coordinate: coord, addressDictionary: nil)
    var mapItem = MKMapItem(placemark: placemark)
    let launchOptions = [MKLaunchOptionsDirectionsModeKey: MKLaunchOptionsDirectionsModeTransit]
    mapItem.openInMapsWithLaunchOptions(launchOptions)
}
```
Launch Maps into Transit Mode

Transit

MKLaunchOptionsDirectionsMode

Sample code

```swift
func openInMapsTransit(coord: CLLocationCoordinate2D) {
    var placemark = MKPlacemark(coordinate: coord, addressDictionary: nil)
    var mapItem = MKMapItem(placemark: placemark)
    let launchOptions = [MKLaunchOptionsDirectionsModeKey: MKLaunchOptionsDirectionsModeTransit]
    mapItem.openInMapsWithLaunchOptions(launchOptions)
}
```
Launch Maps into Transit Mode

Transit

MKLaunchOptionsDirectionsMode

Sample code

```swift
func openInMapsTransit(coord: CLLocationCoordinate2D) {
    var placemark = MKPlacemark(coordinate: coord, addressDictionary: nil)
    var mapItem = MKMapItem(placemark: placemark)
    let launchOptions = [MKLaunchOptionsDirectionsModeKey: MKLaunchOptionsDirectionsModeTransit]
    mapItem.openInMapsWithLaunchOptions(launchOptions)
}
```
Launch Maps into Transit Mode

Transit

MKLaunchOptionsDirectionsMode

Sample code

```swift
func openInMapsTransit(coord: CLLocationCoordinate2D) {
    var placemark = MKPlacemark(coordinate: coord, addressDictionary: nil)
    var mapItem = MKMapItem(placemark: placemark)
    let launchOptions = [MKLaunchOptionsDirectionsModeKey: MKLaunchOptionsDirectionsModeTransit]
    mapItem.openInMapsWithLaunchOptions(launchOptions)
}
```
Demo

New MapKit API
Lesson Learned

New MapKit API
Lesson Learned

New MapKit API

Customize pin annotations
Lesson Learned

New MapKit API

Customize pin annotations
Customize callouts
Lesson Learned

New MapKit API

Customize pin annotations
Customize callouts
Request ETA for Transit
Lesson Learned

New MapKit API

Customize pin annotations
Customize callouts
Request ETA for Transit
Open Maps to provide transit directions
Flyover in MapKit

Elisabeth Lindkvist  Software Engineer Maps Team
Flyover in MapKit
Flyover in MapKit

What is Flyover?
Flyover in MapKit

What is Flyover?
How will Flyover work in your app?
Flyover in MapKit

What is Flyover?
How will Flyover work in your app?
Showing the right region
Flyover in MapKit

What is Flyover?
How will Flyover work in your app?
Showing the right region
Using Flyover with existing API
What is Flyover?
What is Flyover?
What is Flyover?
What is Flyover?

Photorealistic 3D models of cities and landmarks
What is Flyover?

Photorealistic 3D models of cities and landmarks
What is Flyover?

Photorealistic 3D models of cities and landmarks
Available in Maps since iOS 6.0
What is Flyover?

Photorealistic 3D models of cities and landmarks
Available in Maps since iOS 6.0
What is Flyover?
What is Flyover?

Satellite imagery over a 3D terrain model shown where Flyover is not available
What is Flyover?
What is Flyover?

Flyover and satellite imagery is displayed on a globe.
What is Flyover?

Flyover and satellite imagery is displayed on a globe.

Affects region setting and camera handling.
Flyover in Your App
Displaying a Flyover view
Flyover in Your App
Displaying a Flyover view

```swift
enum MKMapType : UInt {
    case Standard
    case Satellite
    case Hybrid
}
```
enum MKMapType : UInt {
    case Standard
    case Satellite
    case Hybrid
}
enum MKMapType : UInt {
    case Standard
    case Satellite
    case Hybrid
}
enum MKMapType : UInt {
    case Standard
    case Satellite
    case Hybrid
}
Flyover in Your App
Displaying a Flyover view

defined as:

```swift
enum MKMapType : UInt {
    case Standard
    case Satellite
    case Hybrid
    case SatelliteFlyover
    case HybridFlyover
}
```
Flyover in Your App

Displaying a Flyover view
Flyover in Your App

Displaying a Flyover view

Use the SatelliteFlyover map type to display Flyover and satellite imagery on terrain
Flyover in Your App
Displaying a Flyover view

Use the SatelliteFlyover map type to display Flyover and satellite imagery on terrain

```swift
mapView.mapType = .SatelliteFlyover
```
Flyover in Your App
Displaying a Flyover view
Flyover in Your App

Displaying a Flyover view

HybridFlyover adds labels, points of interest, roads and borders
Flyover in Your App

Displaying a Flyover view

HybridFlyover adds labels, points of interest, roads and borders

```
mapView.mapType = .HybridFlyover
```
Region Setting in Flyover
Region Setting in Flyover

Rectangular regions

... 

var region: MKCoordinateRegion
var visibleMapRect: MKMapRect

func setRegion(region: MKCoordinateRegion, animated: Bool)
func setVisibleMapRect(mapRect: MKMapRect, animated animate: Bool)

...
Region Setting in Flyover

Rectangular regions

Visible area is not always rectangular
Region Setting in Flyover

Rectangular regions

Visible area is not always rectangular
Region Setting in Flyover

Rectangular regions

Visible area is not always rectangular
Region Setting in Flyover

Rectangular regions

Visible area is not always rectangular
Region Setting in Flyover

Rectangular regions

Visible area is not always rectangular

Visible region defined as enclosing rectangle
Region Setting in Flyover

Rectangular regions

Visible area is not always rectangular

Visible region defined as enclosing rectangle
Region Setting in Flyover
Rectangular regions

Visible area is not always rectangular
Visible region defined as enclosing rectangle
Region Setting in Flyover

Rectangular regions

Visible area is not always rectangular

Visible region defined as enclosing rectangle
Region Setting in Flyover
Rectangular regions

Visible area is not always rectangular
Visible region defined as enclosing rectangle
Larger regions may be wrapped around the globe
Region Setting in Flyover
Using MKMapCamera
Region Setting in Flyover
Using MKMapCamera
Region Setting in Flyover

Using MKMapCamera

Four basic properties
Region Setting in Flyover
Using MKMapCamera

Four basic properties

```swift
var centerCoordinate: CLLocationCoordinate2D
```
Region Setting in Flyover
Using MKMapCamera

Four basic properties

```swift
var centerCoordinate: CLLocationCoordinate2D
var heading: CLLocationDirection
```
Region Setting in Flyover
Using MKMapCamera

Four basic properties

```swift
var centerCoordinate: CLLocationCoordinate2D
var heading: CLLocationDirection
var pitch: CGFloat
```
Region Setting in Flyover
Using MKMapCamera

Four basic properties

```swift
var centerCoordinate: CLLocationCoordinate2D
var heading: CLLocationDirection
var pitch: CGFloat
var altitude: CLLocationDistance
```
Region Setting in Flyover
Using MKMapCamera

convenience init(lookingAtCenterCoordinate
    centerCoordinate: CLLocationCoordinate2D,
    fromEyeCoordinate eyeCoordinate: CLLocationCoordinate2D,
    eyeAltitude: CLLocationDistance)
Region Setting in Flyover
Using MKMapCamera

costom convenience init(lookingAtCenterCoordinate
centerCoordinate: CLLocationCoordinate2D,
fromEyeCoordinate eyeCoordinate: CLLocationCoordinate2D,
eyeAltitude: CLLocationDistance)
Region Setting in Flyover
Using MKMapCamera

c convenience init(lookingAtCenterCoordinate
centerCoordinate: CLLocationCoordinate2D,
fromEyeCoordinate eyeCoordinate: CLLocationCoordinate2D,
eyeAltitude: CLLocationDistance)
Region Setting in Flyover
Using MKMapCamera

cellinenst init(lookingAtCenterCoordinate
centerCoordinate: CLLocationCoordinate2D,
fromEyeCoordinate eyeCoordinate: CLLocationCoordinate2D,
eyeAltitude: CLLocationDistance)
Region Setting in Flyover

Using MKMapCamera

Terrain and buildings in Flyover makes **distance** between the camera and center coordinate a better measure than altitude
Region Setting in Flyover
Using MKMapCamera

convenience init(lookingAtCenterCoordinate
centerCoordinate: CLLocationCoordinate2D,
fromDistance distance: CLLocationDistance,
pitch: CGFloat,
heading: CLLocationDirection)
Region Setting in Flyover
Using MKMapCamera

convenience
init(lookingAtCenterCoordinate
centerCoordinate: CLLocationCoordinate2D,
fromDistance distance: CLLocationCoordinateDistance,
pitch: CGFloat,
heading: CLLocationCoordinateDirection)
Region Setting in Flyover
Using MKMapCamera

convenience
init(lookingAtCenterCoordinate
centerCoordinate:
CLLocationCoordinate2D,
fromDistance distance:
CLLocationDistance,
pitch: CGFloat,
heading: CLLocationDirection)
Region Setting in Flyover
Using MKMapCamera

convenience
init(lookingAtCenterCoordinate
centerCoordinate: 
CLLocationCoordinate2D,
fromDistance distance: 
CLLocationDistance,
pitch: CGFloat,
heading: CLLocationCoordinateDirection)
Region Setting in Flyover
Using MKMapCamera

convenience
init(lookingAtCenterCoordinate
centerCoordinate: 
CLLocationCoordinate2D,
fromDistance distance: 
CLLocationDistance,
pitch: CGFloat,
heading: CLLocationDirection)
Region Setting in Flyover

Using MKMapCamera

convenience
init(lookingAtCenterCoordinate
centerCoordinate:
CLLocationCoordinate2D,
fromDistance distance:
CLLocationDistance,
pitch: CGFloat,
heading: CLLocationDirection)
Demo
Looking around in Flyover
Flyover and Existing API
Flyover and Existing API

MKAnnotation
Flyover and Existing API

MKAnnotation

Use to annotate single coordinate
Flyover and Existing API

MKAnnotation

Use to annotate single coordinate

• Placed on top of 3D buildings on Standard map type
Flyover and Existing API

MKAnnotation

Use to annotate single coordinate
• Placed on top of 3D buildings on Standard map type
Flyover and Existing API

MKAnnotation

Use to annotate single coordinate
- Placed on top of 3D buildings on Standard map type
- Placed on top of buildings and terrain on SatelliteFlyover and HybridFlyover map types
Flyover and Existing API

MKOverlay
Flyover and Existing API

MKOverlay

Use overlays to highlight areas of the map
Flyover and Existing API

MKOverlay

Use overlays to highlight areas of the map
• Occluded by 3D buildings on Standard map type
Flyover and Existing API
MKOverlay

Use overlays to highlight areas of the map
• Occluded by 3D buildings on Standard map type
Flyover and Existing API

MKOverlay

Use overlays to highlight areas of the map

- Occluded by 3D buildings on Standard map type
- Occluded by Flyover buildings and trees
Flyover and Existing API

MKOverlay

Use overlays to highlight areas of the map

• Occluded by 3D buildings on Standard map type
• Occluded by Flyover buildings and trees
Flyover and Existing API

MKOverlay

Use overlays to highlight areas of the map

• Occluded by 3D buildings on Standard map type
• Occluded by Flyover buildings and trees
Flyover and Existing API

MKOverlay

Use overlays to highlight areas of the map

• Occluded by 3D buildings on Standard map type
• Occluded by Flyover buildings and trees
• Drawn on top of terrain
Flyover and Existing API

MKOverlay

Use overlays to highlight areas of the map
- Occluded by 3D buildings on Standard map type
- Occluded by Flyover buildings and trees
- Drawn on top of terrain
Summary
Summary

New ways to customize pins, callouts and map display
Summary

New ways to customize pins, callouts and map display
Transit ETA requests and Maps launch option
Summary

New ways to customize pins, callouts, and map display

Transit ETA requests and Maps launch option

Use SatelliteFlyover and HybridFlyover map types to show photorealistic 3D models and satellite imagery on terrain
More Information

Documentation
MapKit
http://developer.apple.com/maps

Technical Support
Apple Developer Forum
http://developer.apple.com/forums

General Inquiries
Craig Keithley, Technology Evangelist
keithly@apple.com
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>What’s New in Core Location</td>
<td>Pacific Heights</td>
<td>Thursday 1:30PM</td>
</tr>
<tr>
<td>What’s New in Core Audio</td>
<td>Nob Hill</td>
<td>Wednesday 4:30PM</td>
</tr>
<tr>
<td>Lab</td>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>MapKit Lab</td>
<td>Wednesday 11:00AM</td>
<td></td>
</tr>
<tr>
<td>Frameworks Lab E</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Apple WWDC 2015