

```
/*
Copyright (C) 2016 Apple Inc. All Rights
Reserved.
See LICENSE.txt for this sample's licensing
information

Abstract:
Controls the map view and manages the reverse
geocoder to get the current address.
*/
```

```
#import "MapViewController.h"
#import "PlacemarkViewController.h"
```

```
@interface MapViewController () <
    CLLocationManagerDelegate>
```

```
@property (nonatomic, weak) IBOutlet MKMapView *
    mapView;
```

```
@property (nonatomic, weak) IBOutlet
    UIBarButtonItem *getAddressButton;
```

```
@property (nonatomic, strong) CLLocationManager
    *locationManager;
```

```
@property (nonatomic, strong) CLGeocoder *
    geocoder;
```

```
@property (nonatomic, strong) MKPlacemark *
    placemark;
```

```
@end
```

```
@implementation MapViewController
```

```
- (void)viewDidLoad
{
```

```
[super viewDidLoad];

self.locationManager = [[CLLocationManager
    alloc] init];
self.locationManager.delegate = self;

// Gets user permission use location while
// the app is in the foreground.
[self.locationManager
    requestWhenInUseAuthorization];

self.geocoder = [[CLGeocoder alloc] init];
}

- (void)prepareForSegue:(UIStoryboardSegue *)
    segue sender:(id)sender
{
    if ([segue.identifier
        isEqualToString:@"pushToDetail"])
    {
        // Get the destination view controller
        // and set the placemark data that it
        // should display.
        PlacemarkViewController *viewController
            = segue.destinationViewController;
        viewController.placemark = self.
            placemark;
    }
}

- (void)mapView:(MKMapView *)mapView
    didUpdateUserLocation:(MKUserLocation *)
    userLocation
{

```

```
// Center the map the first time we get a
    real location change.
static dispatch_once_t centerMapFirstTime;

if ((userLocation.coordinate.latitude != 0.0
    ) && (userLocation.coordinate.longitude
    != 0.0)) {
    dispatch_once(&centerMapFirstTime, ^{
        [self.mapView setCenterCoordinate:
            userLocation.coordinate
            animated:YES];
    });
}

// Lookup the information for the current
    location of the user.
[self.geocoder reverseGeocodeLocation:self.
    mapView.userLocation.location
    completionHandler:^(NSArray *placemarks,
    NSError *error) {
    if ((placemarks != nil) && (placemarks.
        count > 0)) {
        // If the placemark is not nil then
            we have at least one placemark.
            Typically there will only be
            one.
        self.placemark = placemarks[0];

        // we have received our current
            location, so enable the "Get
            Current Address" button
        self.getAddressButton.enabled = YES;
    }
    else {
```

```
        // Handle the nil case if necessary.
    }
    }];
}

- (void)mapView:(MKMapView *)mapView
  didFailToLocateUserWithError:(NSError *)
  error {

    self.getAddressButton.enabled = NO;

    if (!self.presentedViewController) {
        NSString *message = nil;
        if (error.code ==
            kCLErrorLocationUnknown) {
            // If you receive this error while
            // using the iOS Simulator,
            // location simulation may not be
            // on. Choose a location from the
            // Debug > Simulate Location menu
            // in Xcode.
            message = @"Your location could not
                be determined.";
        }
        else {
            message = error.localizedDescription
                ;
        }

        UIAlertController *alert =
            [UIAlertController
             alertControllerWithTitle:@"Error"

                                     message:message
```

```
                                preferredStyle:
                                UIAlertController
                                styleAlert];
[alert addAction:[UIAlertAction
    actionWithTitle:@"OK"

                                style:
                                UIAlertActionSt
                                yleDefault

                                handler:nil]];
[self presentViewController:alert
    animated:YES completion:nil];
}
}
```

```
- (void)locationManager:(CLLocationManager *)
    manager didChangeAuthorizationStatus:
    (CLAuthorizationStatus)status {
    if (status ==
        kCLAuthorizationStatusRestricted ||
        status == kCLAuthorizationStatusDenied)
    {
        UIAlertController *alert =
            [UIAlertController
                alertControllerWithTitle:@"Location
                Disabled"

                                message:@"Please
                                enable
                                location
                                services in the
                                Settings app."

```

```
                                preferredStyle:
                                UIAlertControllerStyleAlert];
[alert addAction:[UIAlertAction
    actionWithTitle:@"OK"

                                style:
                                UIAlertActionStyleDefault

                                handler:nil]];
[self presentViewController:alert
    animated:YES completion:nil];
}
else if (status ==
    kCLAuthorizationStatusAuthorizedWhenInUse) {
    // This will implicitly try to get the
    // user's location, so this can't be
    // set
    // until we know the user granted this
    // app location access
    self.mapView.showsUserLocation = YES;
}
}

@end
```