

What's New in iOS Notifications

Session 713

Rhett Dickson

Software Engineer

Dan Kurtz

Software Engineer

Previously in iOS Notifications

User notifications

Silent notifications

User Notifications

What



User Notifications

What



User Notifications

What



User Notifications

What



User Notifications

Where



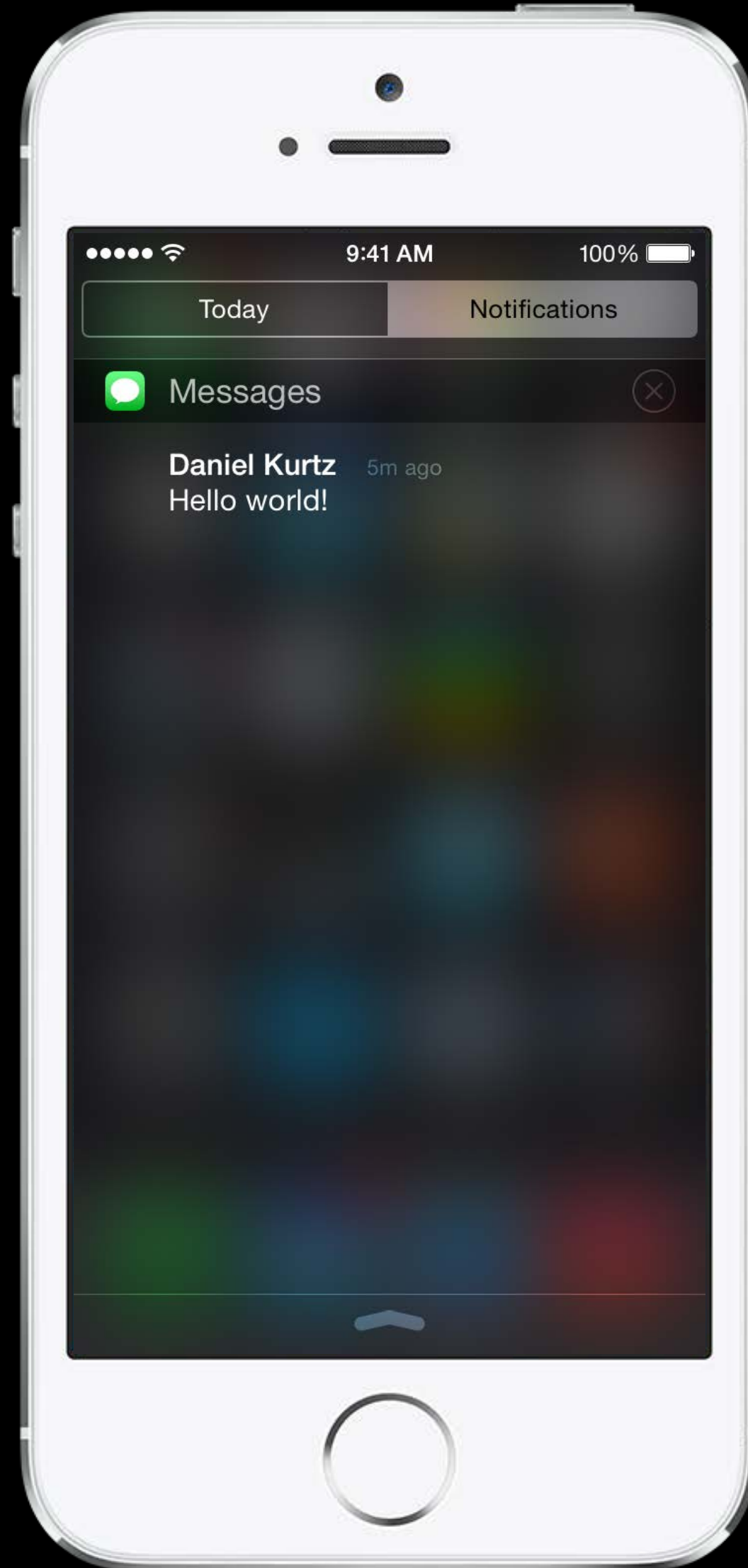
User Notifications

Where



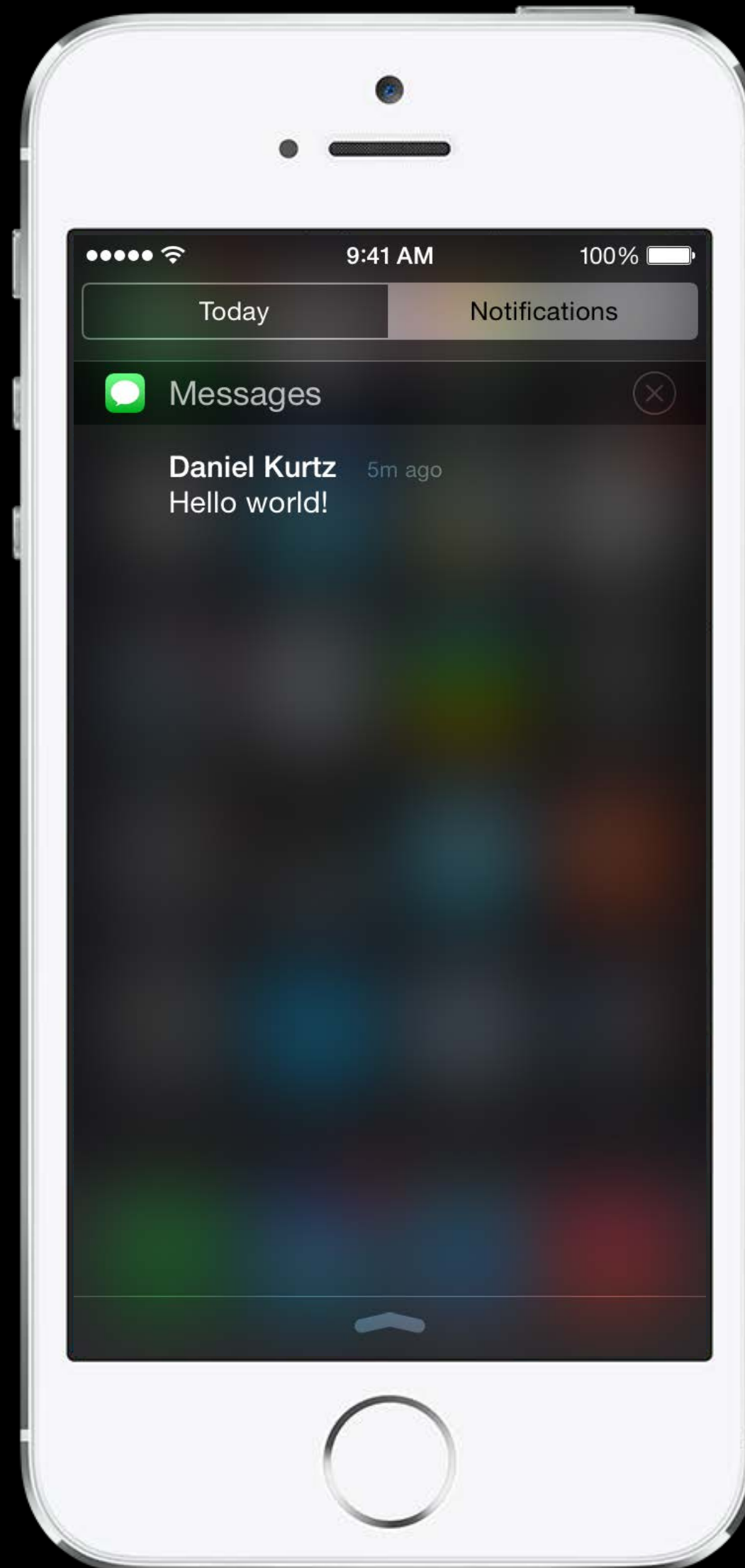
User Notifications

Where



User Notifications

Where



Where Notifications Come From



App

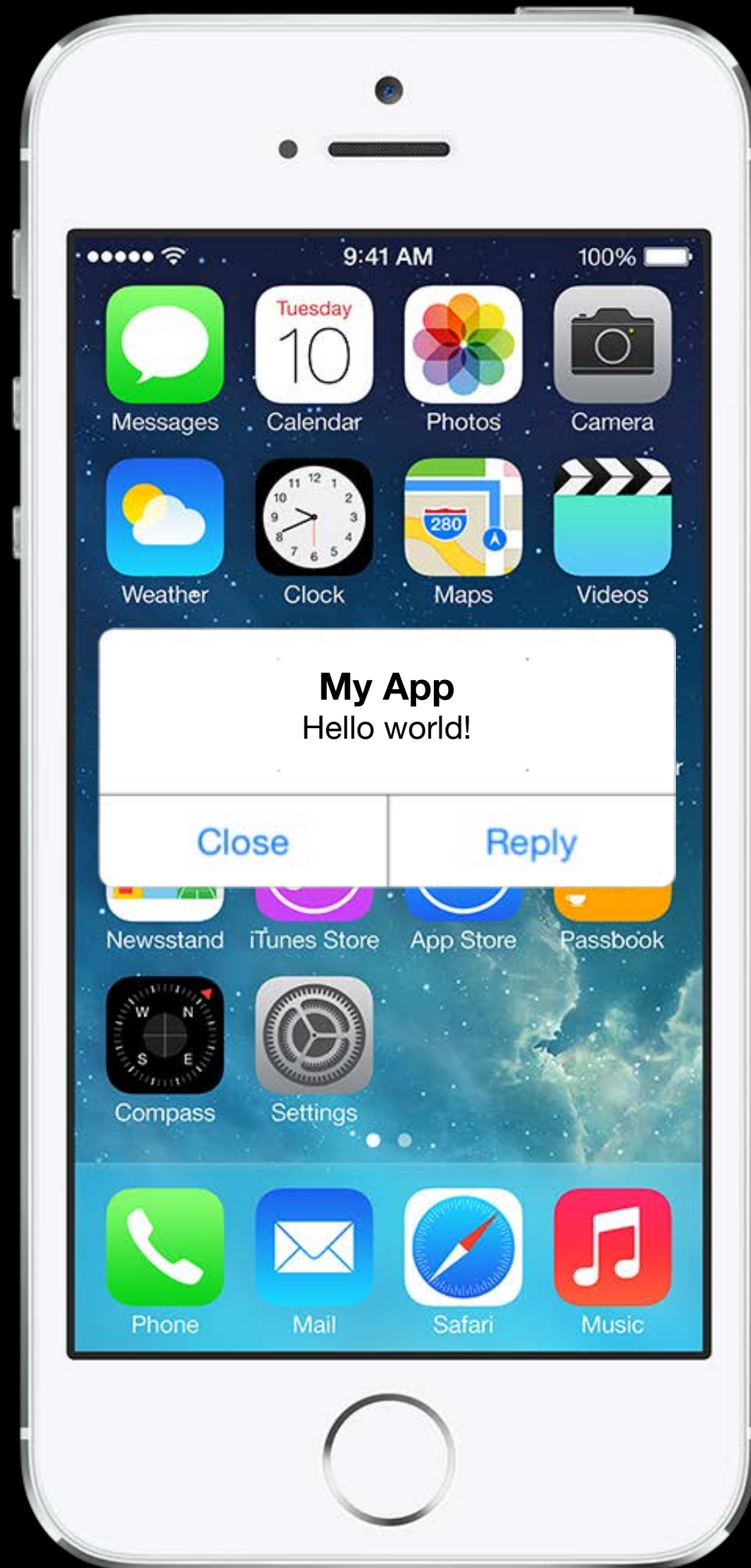


APNs

Where Notifications Come From



App



APNs

Where Notifications Come From



App



APNs

Where Notifications Come From



App



APNs



```
aps {  
  alert: {...}  
}
```

Where Notifications Come From



App

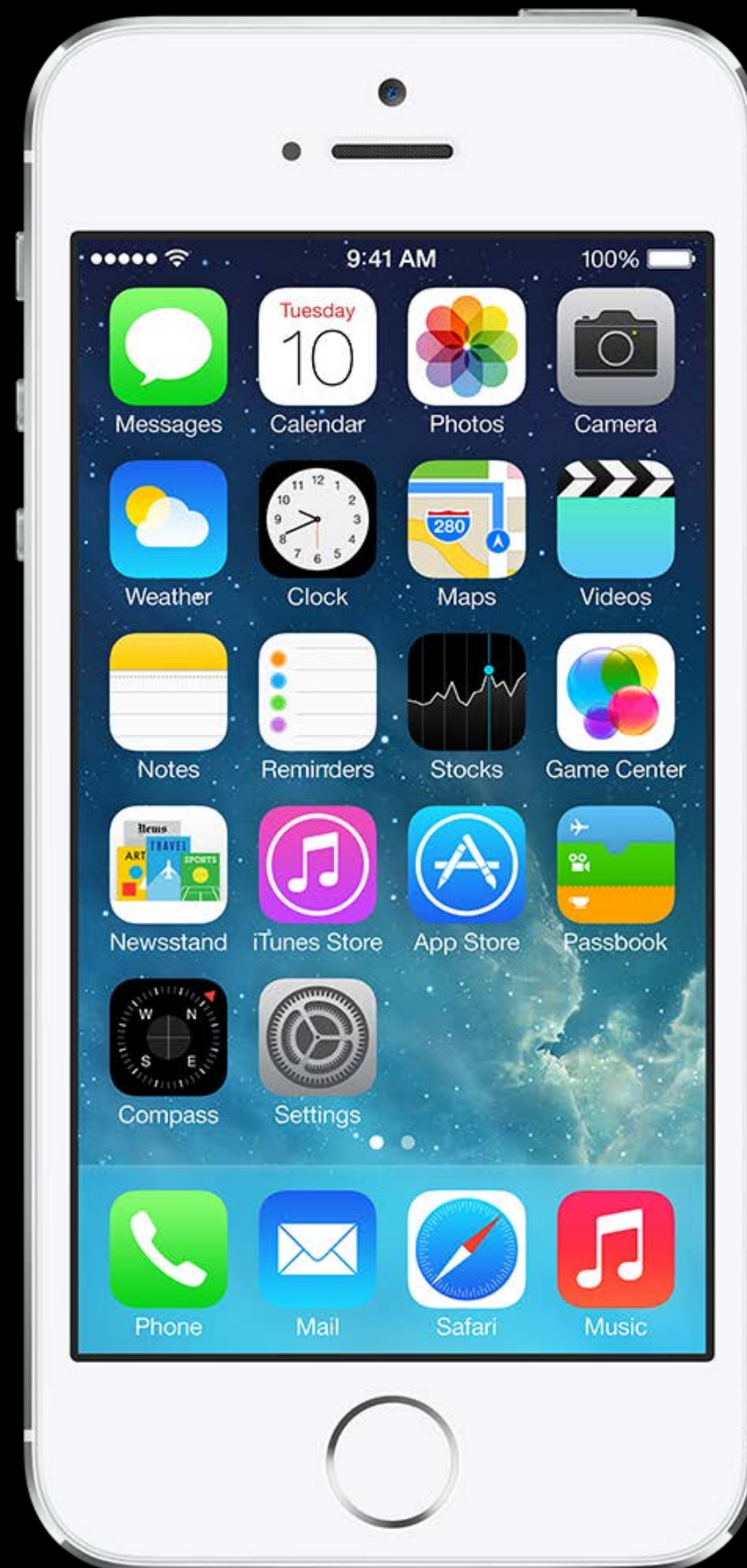


APNs

Where Notifications Come From



App



APNs

Where Notifications Come From



App



APNs

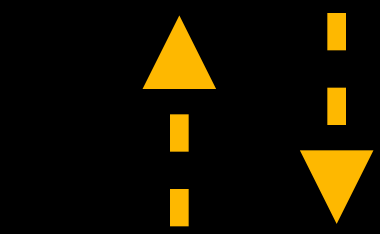


```
aps {  
  content-available: 1  
}
```

Where Notifications Come From



Content Server



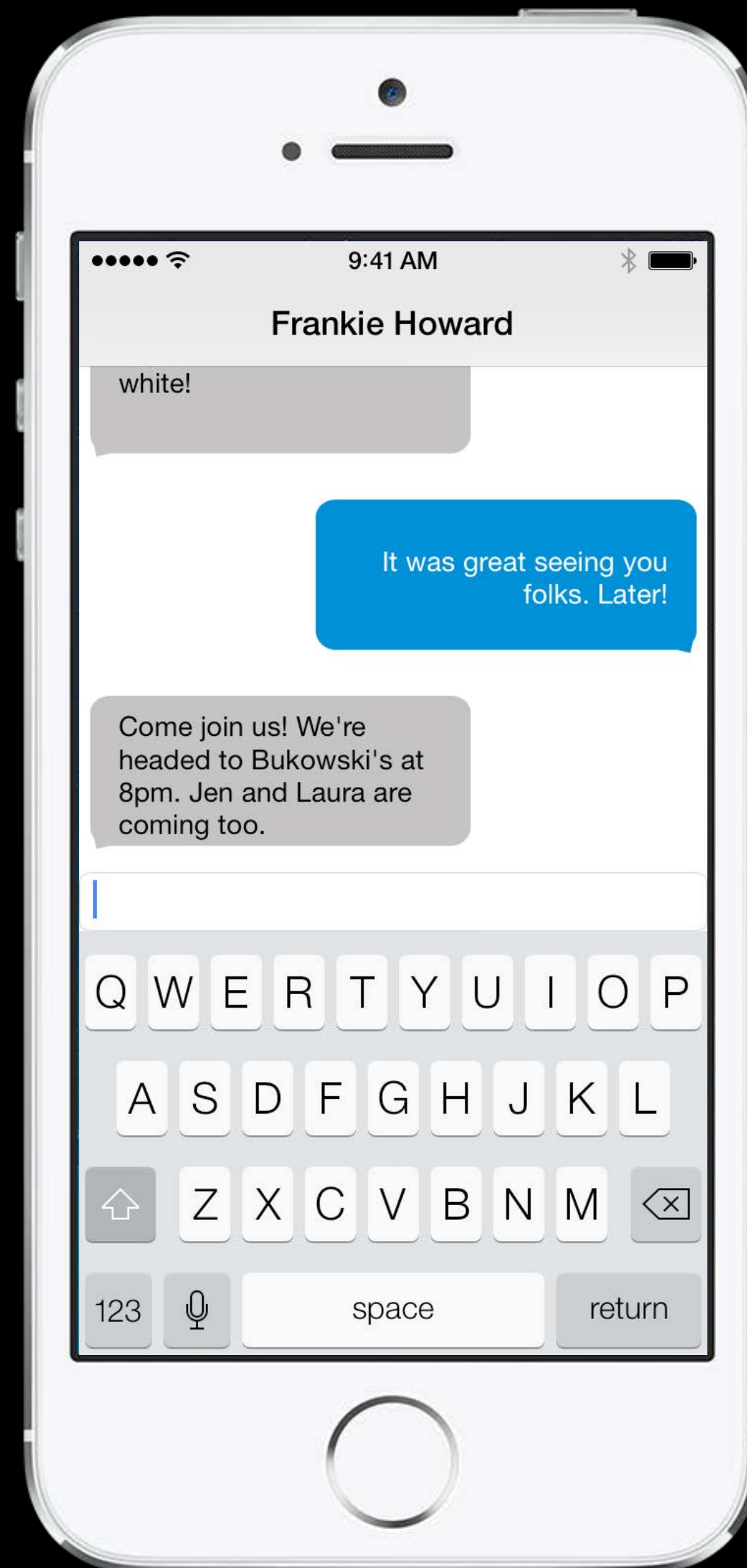
App



APNs

```
aps {  
  content-available: 1  
}
```

Where Notifications Come From



Today



User Notifications

Notification Actions

Remote Notifications

Location Notifications

User Notifications

User Notifications

What's new

Must register to use

Require user approval



User Notifications

Registering

```
UIUserNotificationType types = UIUserNotificationTypeBadge |  
    UIUserNotificationTypeSound | UIUserNotificationTypeAlert;  
  
UIUserNotificationSettings *mySettings = [UIUserNotificationSettings  
    settingsForTypes:types categories:nil];  
  
[[UIApplication sharedApplication]  
    registerUserNotificationSettings:mySettings];
```

User Notifications

Registering

```
UIUserNotificationType types = UIUserNotificationTypeBadge |  
    UIUserNotificationTypeSound | UIUserNotificationTypeAlert;
```

```
UIUserNotificationSettings *mySettings = [UIUserNotificationSettings  
    settingsForTypes:types categories:nil];
```

```
[[UIApplication sharedApplication]  
    registerUserNotificationSettings:mySettings];
```


User Notifications

Registering

```
UIUserNotificationType types = UIUserNotificationTypeBadge |  
    UIUserNotificationTypeSound | UIUserNotificationTypeAlert;
```

```
UIUserNotificationSettings *mySettings = [UIUserNotificationSettings  
    settingsForTypes:types categories:nil];
```

```
[[UIApplication sharedApplication]  
    registerUserNotificationSettings:mySettings];
```

User Notifications

Registering

```
UIUserNotificationType types = UIUserNotificationTypeBadge |  
    UIUserNotificationTypeSound | UIUserNotificationTypeAlert;
```

```
UIUserNotificationSettings *mySettings = [UIUserNotificationSettings  
    settingsForTypes:types categories:nil];
```

```
[[UIApplication sharedApplication]  
    registerUserNotificationSettings:mySettings];
```

User Notifications

Registering

```
UIUserNotificationType types = UIUserNotificationTypeBadge |  
    UIUserNotificationTypeSound | UIUserNotificationTypeAlert;  
  
UIUserNotificationSettings *mySettings = [UIUserNotificationSettings  
    settingsForTypes:types categories:nil];  
  
[[UIApplication sharedApplication]  
    registerUserNotificationSettings:mySettings];
```

User Notifications

Registering

```
UIUserNotificationType types = UIUserNotificationTypeBadge |  
    UIUserNotificationTypeSound | UIUserNotificationTypeAlert;  
  
UIUserNotificationSettings *mySettings = [UIUserNotificationSettings  
    settingsForTypes:types categories:nil];  
  
[[UIApplication sharedApplication]  
    registerUserNotificationSettings:mySettings];
```

“My App” Would Like to Send You Notifications

Notifications may include alerts,
sounds, and icon badges. These can
be configured in Settings.

Don't Allow

OK

User Notifications

UIApplicationDelegate Callback

```
- (void)application:(UIApplication *)application
didRegisterUserNotificationSettings:
(UIUserNotificationSettings *)notificationSettings {

    // user has allowed receiving user notifications of the following types
    UIUserNotificationType allowedTypes = [notificationSettings types];

}
```

User Notifications

UIApplicationDelegate Callback

```
- (void)application:(UIApplication *)application  
didRegisterUserNotificationSettings:  
(UIUserNotificationSettings *)notificationSettings {
```

```
// user has allowed receiving user notifications of the following types  
UIUserNotificationType allowedTypes = [notificationSettings types];
```

```
}
```

User Notifications

UIApplicationDelegate Callback

```
- (void)application:(UIApplication *)application
didRegisterUserNotificationSettings:
(UIUserNotificationSettings *)notificationSettings {

    // user has allowed receiving user notifications of the following types
    UIUserNotificationType allowedTypes = [notificationSettings types];

}
```

User Notifications

Getting Settings

```
- (void)getReadyForNotification {  
    // ...  
  
    // check to make sure we still need to show notification  
    UIUserNotificationSettings *currentSettings = [[UIApplication  
        sharedApplication] currentUserNotificationSettings];  
  
    [self checkSettings:currentSettings];  
}
```


User Notifications

Getting Settings

```
- (void)getReadyForNotification {  
    // ...  
  
    // check to make sure we still need to show notification  
    UIUserNotificationSettings *currentSettings = [[UIApplication  
        sharedApplication] currentUserNotificationSettings];  
  
    [self checkSettings:currentSettings];  
}
```

User Notifications

```
-[UIApplication registerUserNotificationSettings:]  
-[UIApplication currentNotificationSettings]  
+[UIUserNotificationSettings settingsForTypes:categories:]
```

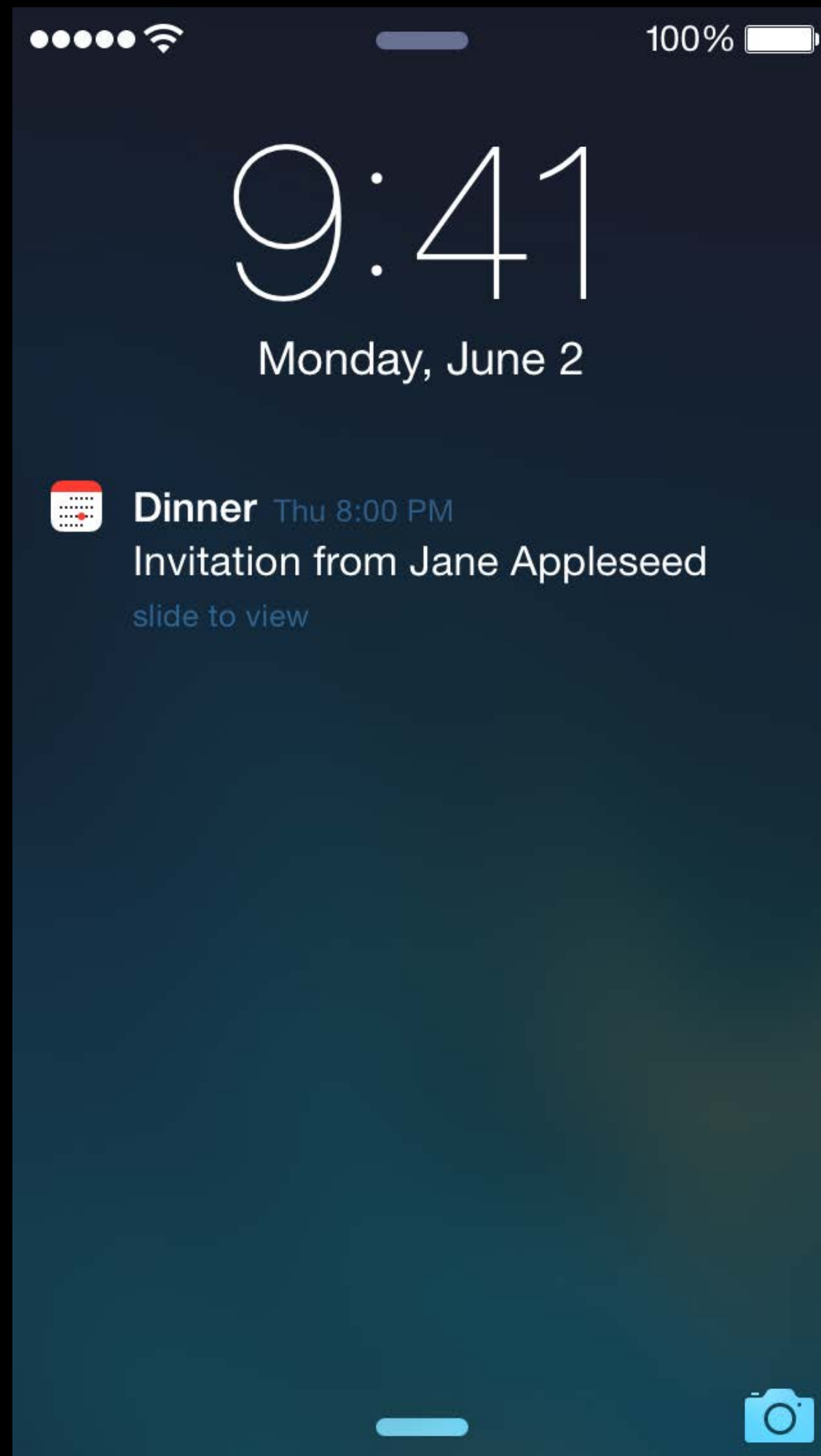
Notification Actions

Notification Actions

In iOS 7

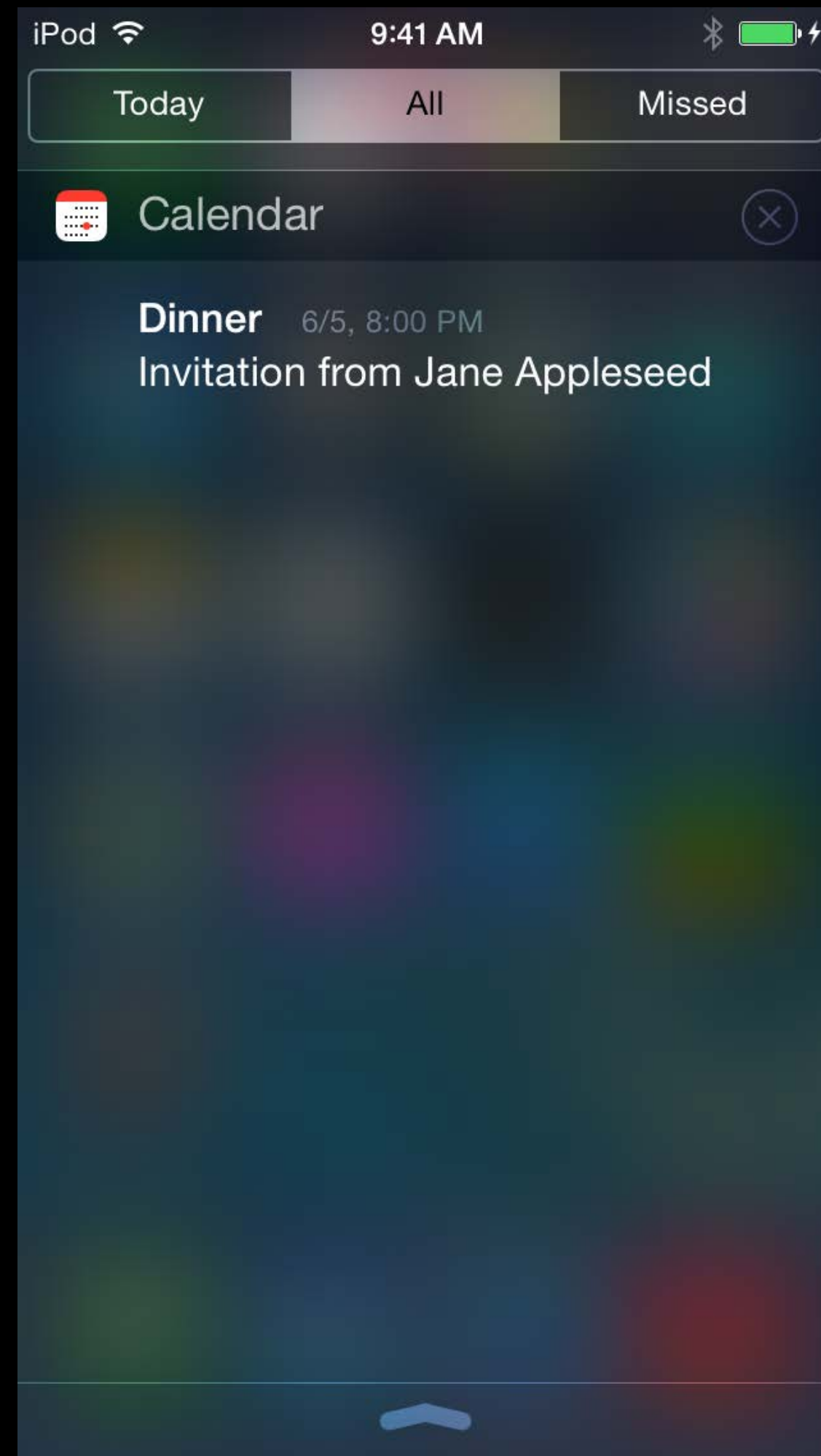
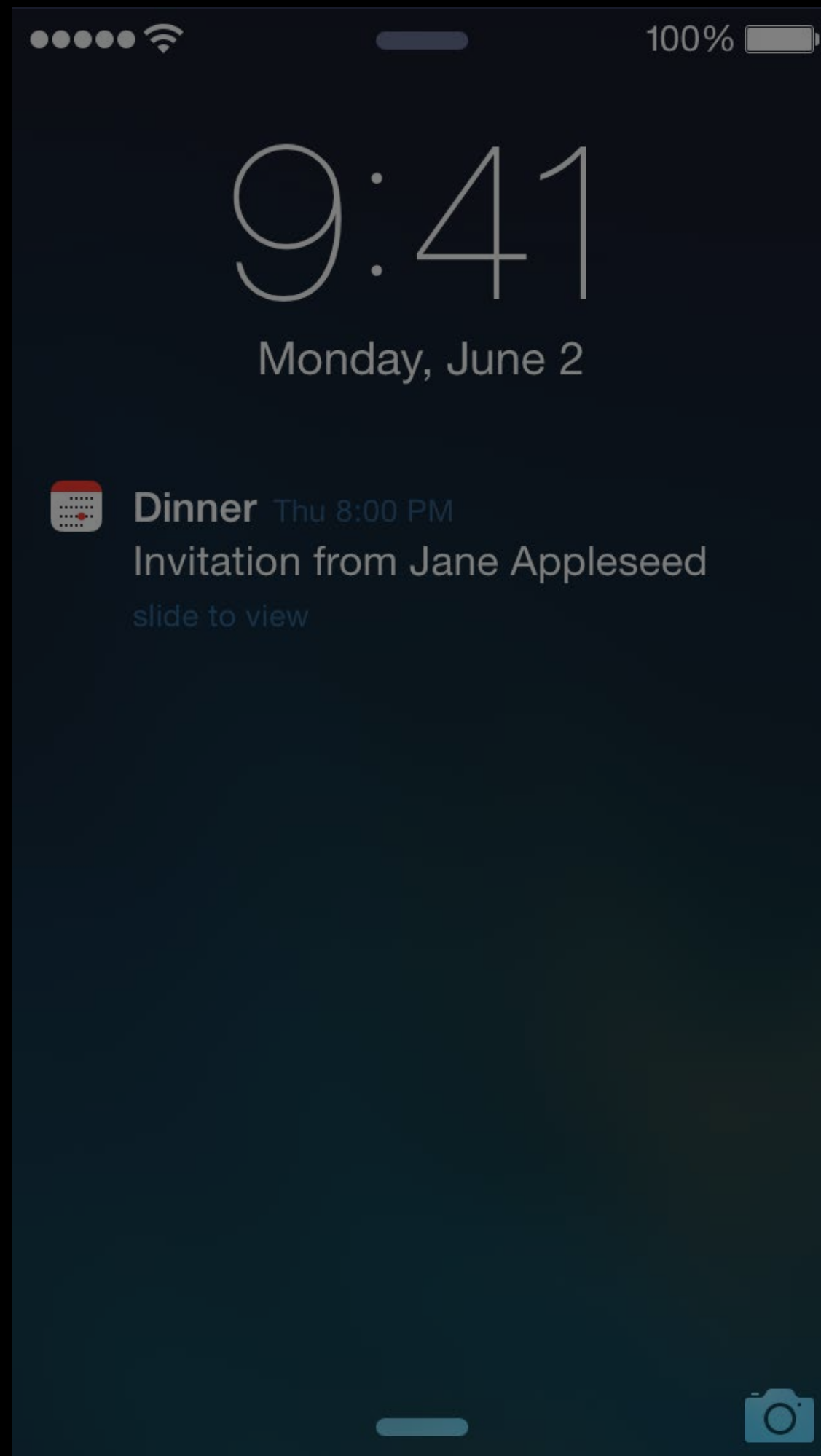
Notification Actions

In iOS 7



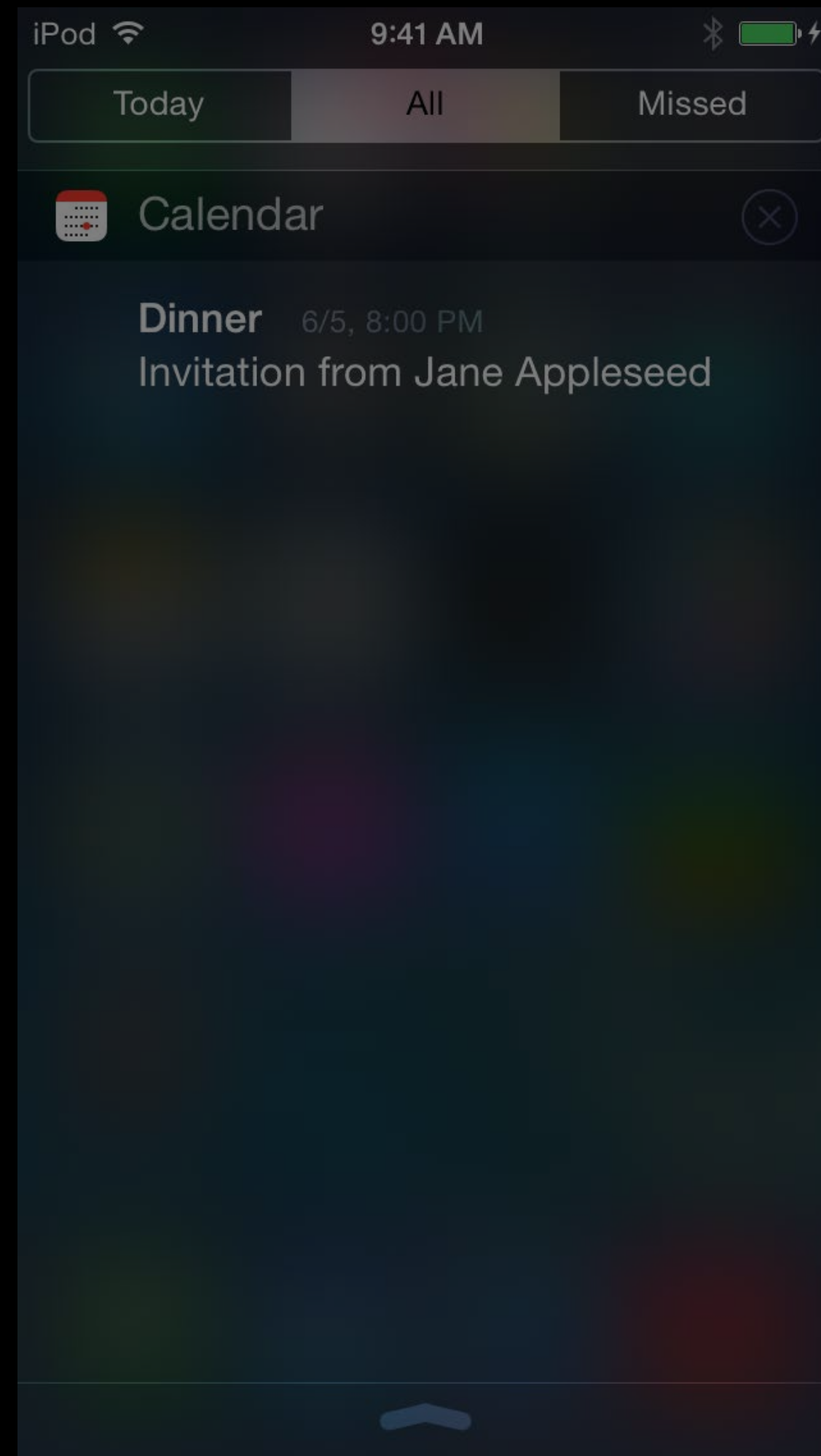
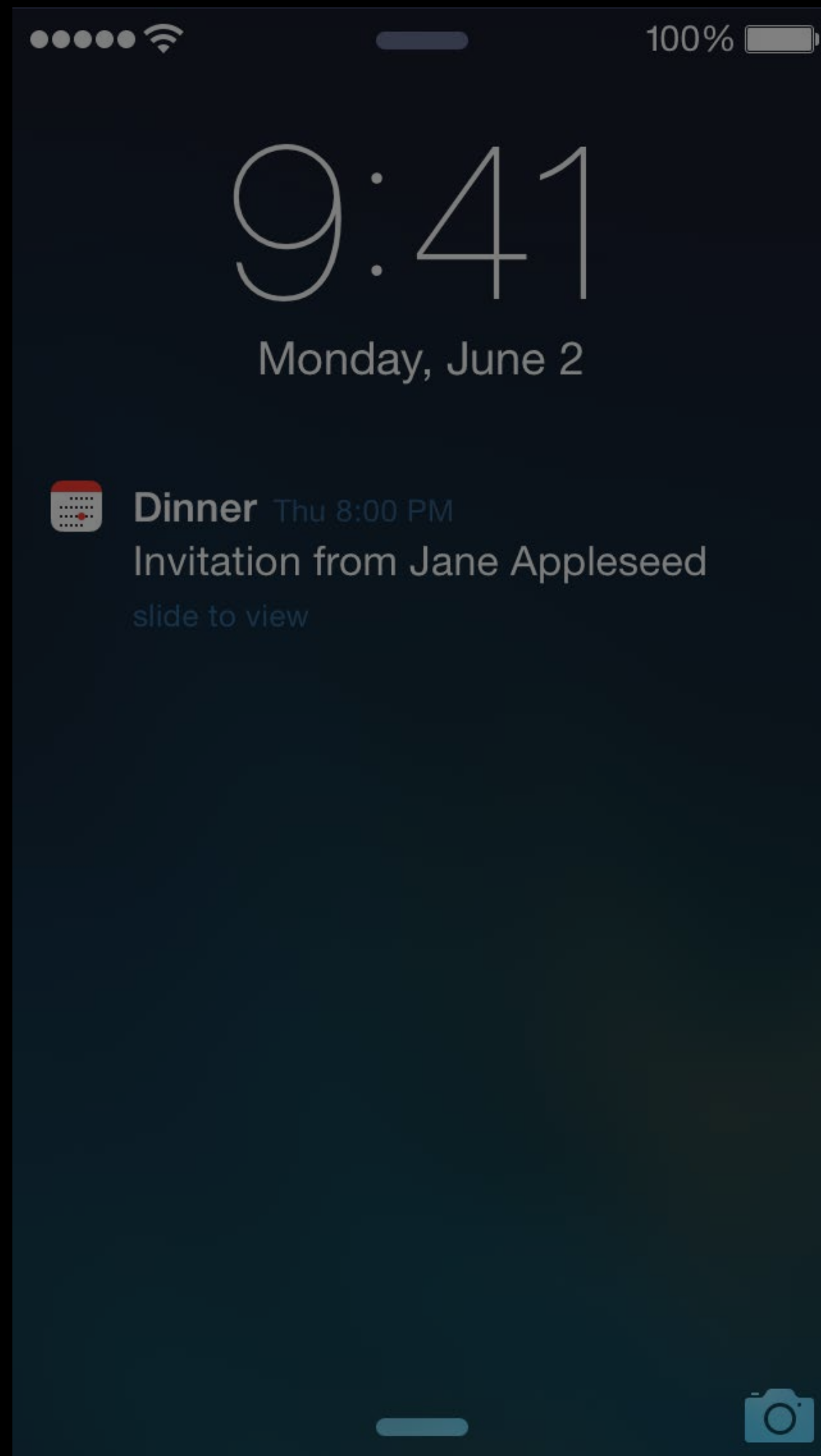
Notification Actions

In iOS 7



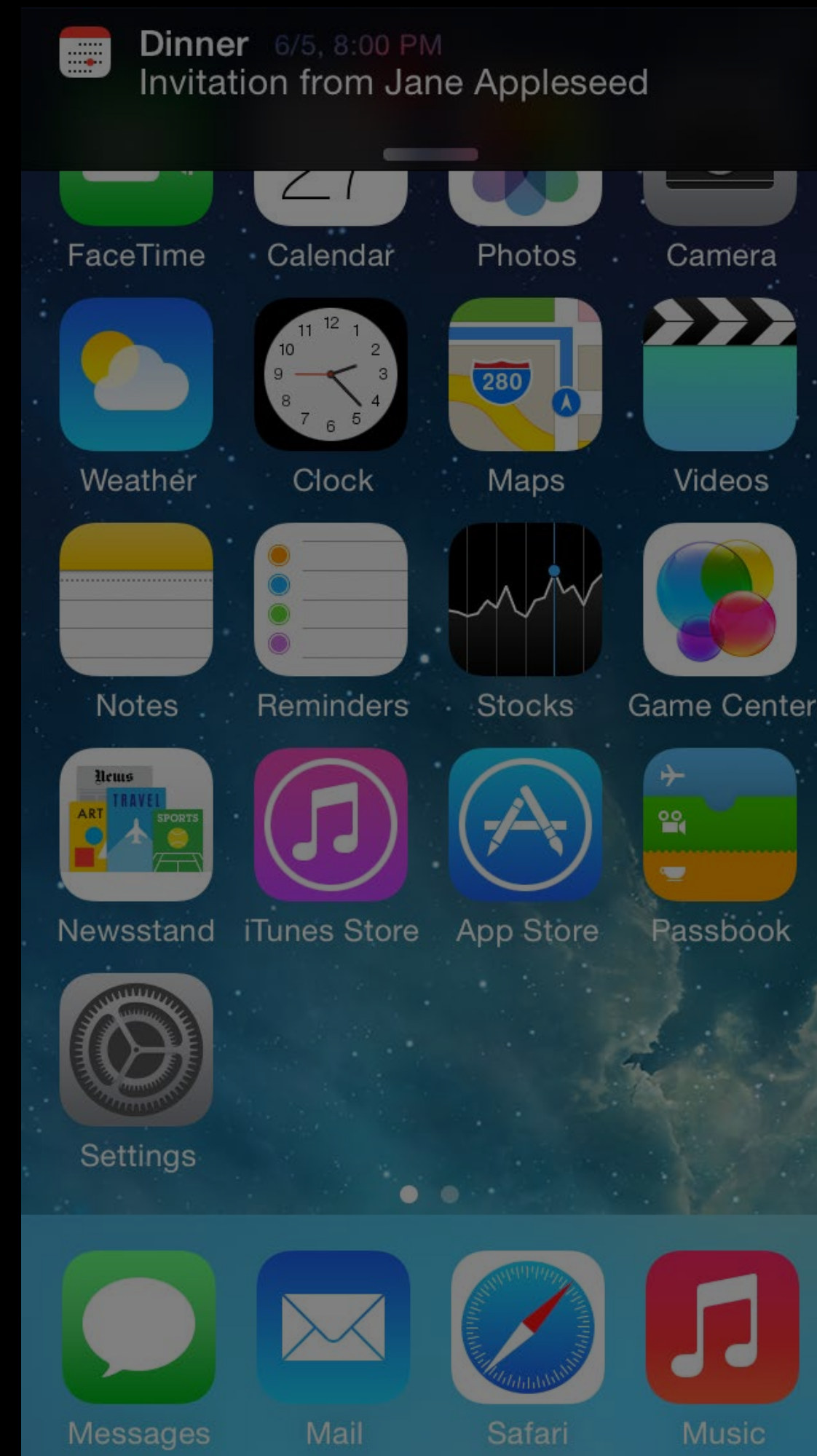
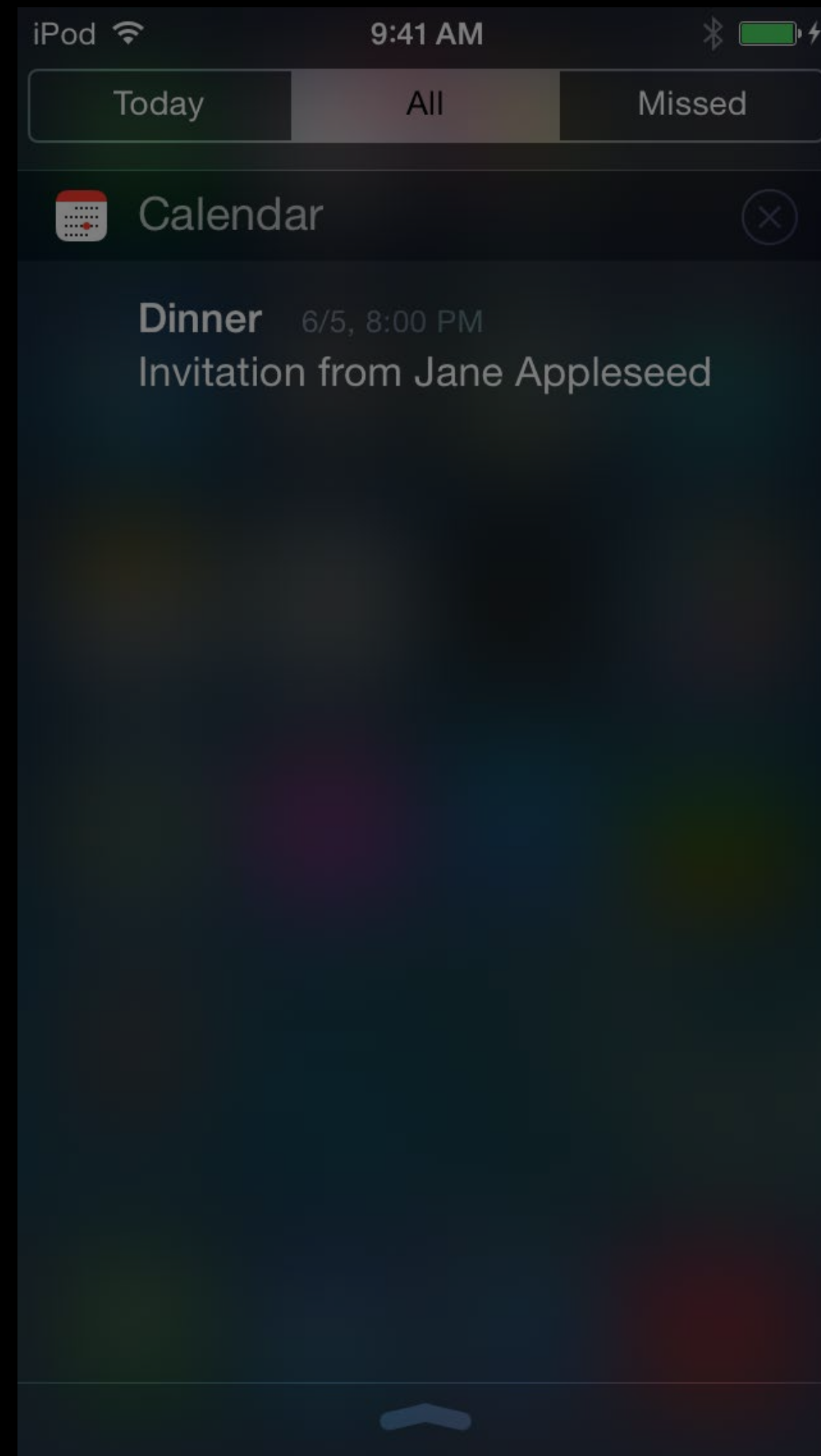
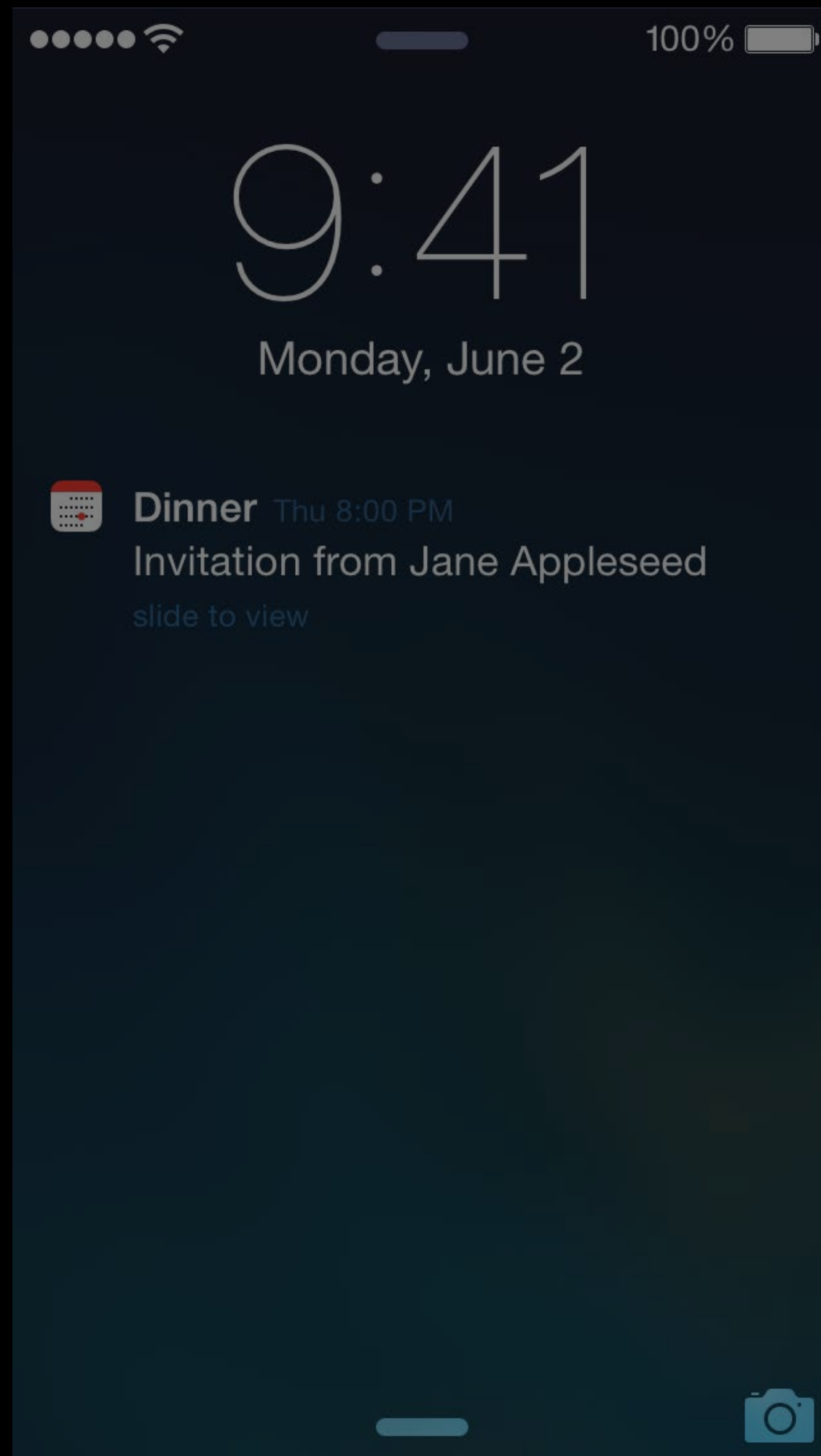
Notification Actions

In iOS 7



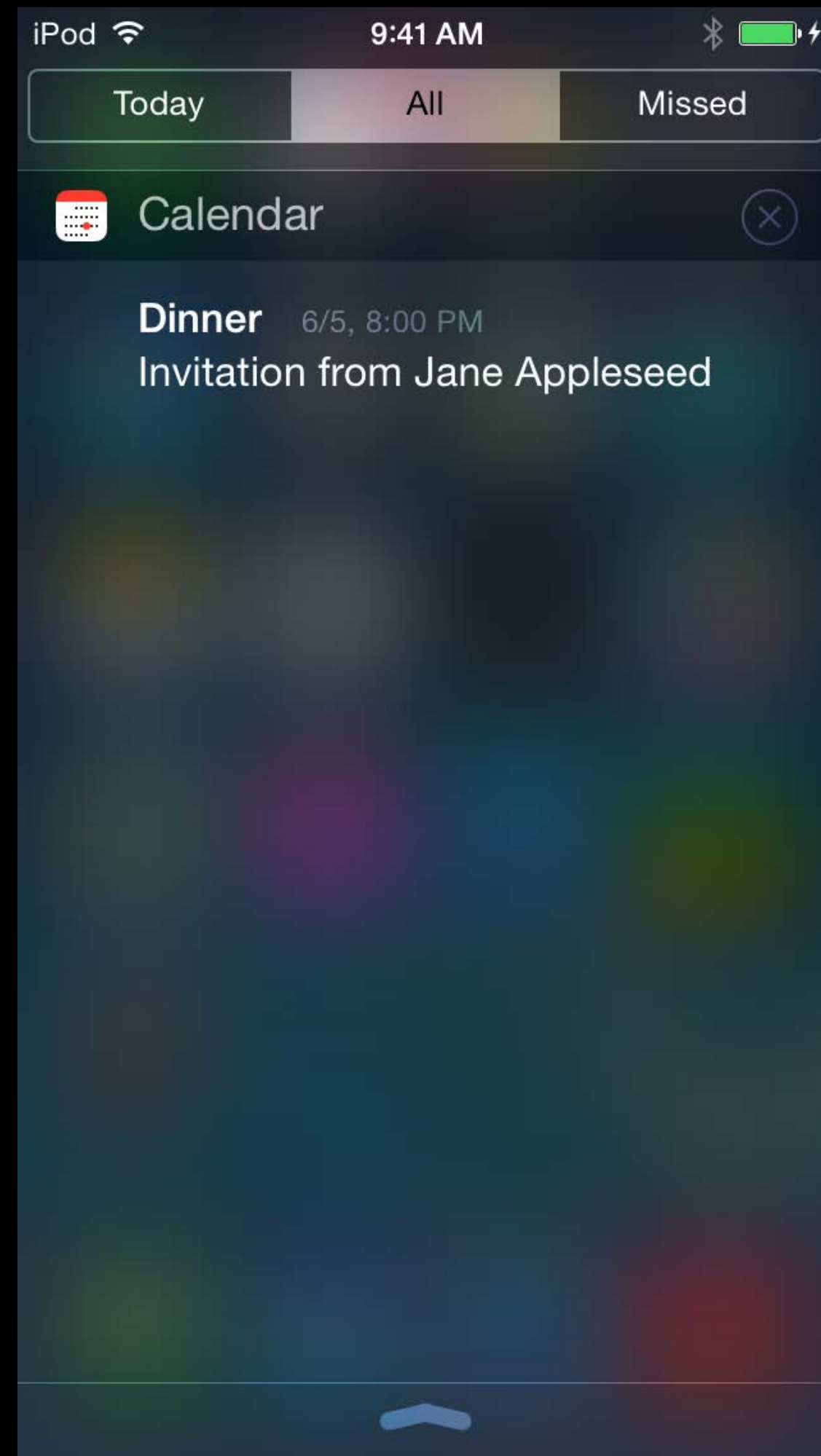
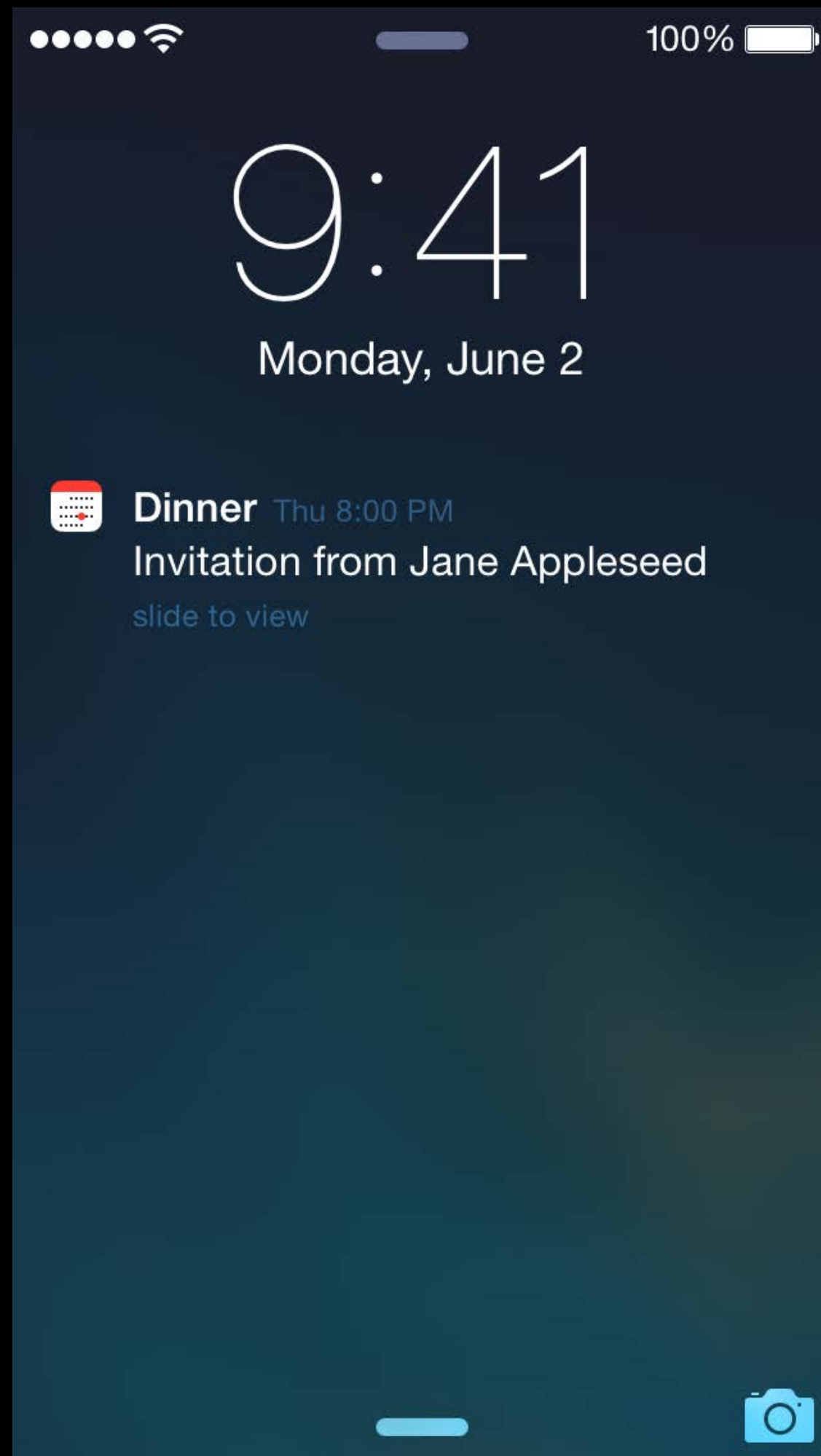
Notification Actions

In iOS 7



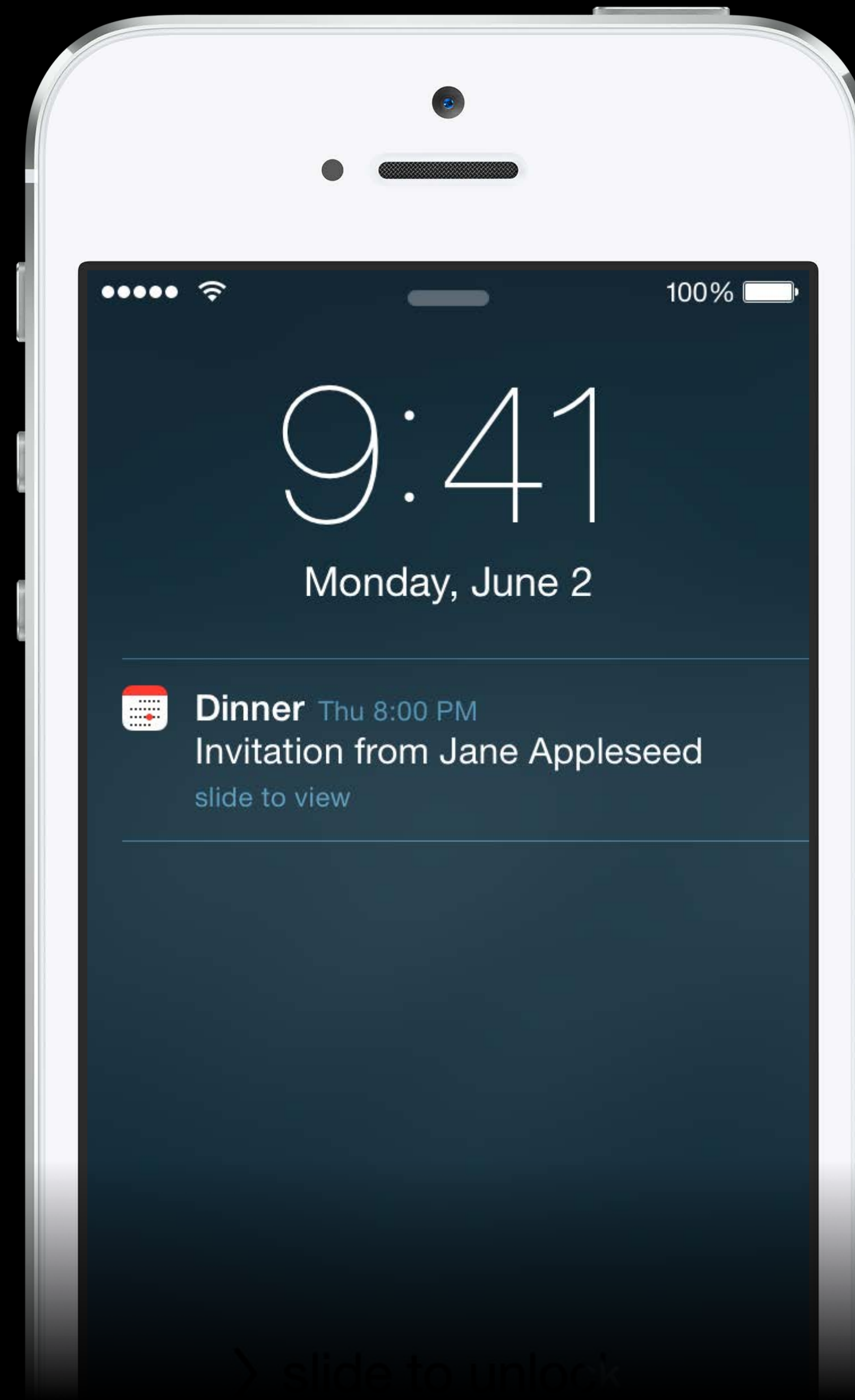
Notification Actions

In iOS 7



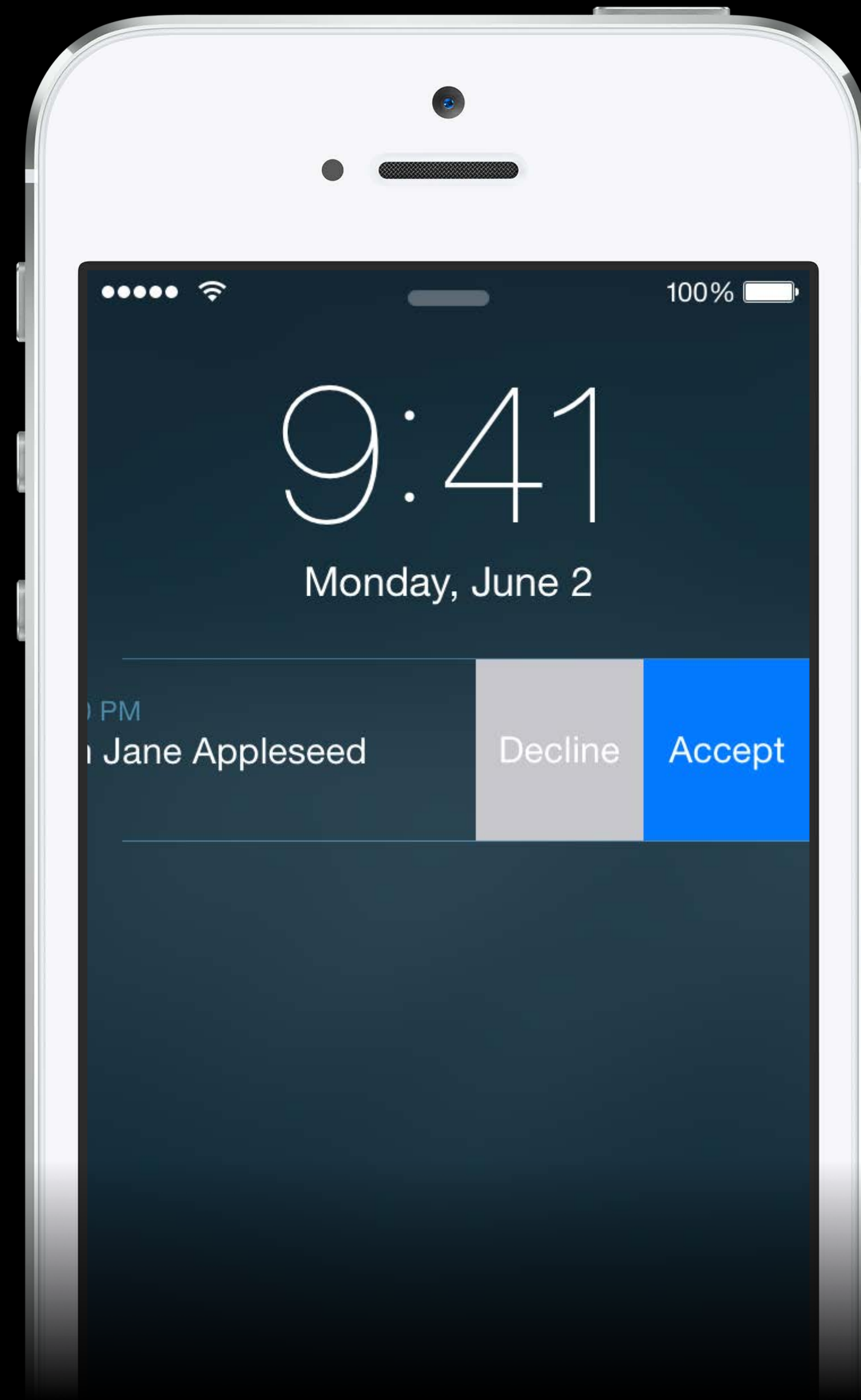
Notification Actions

In iOS 8



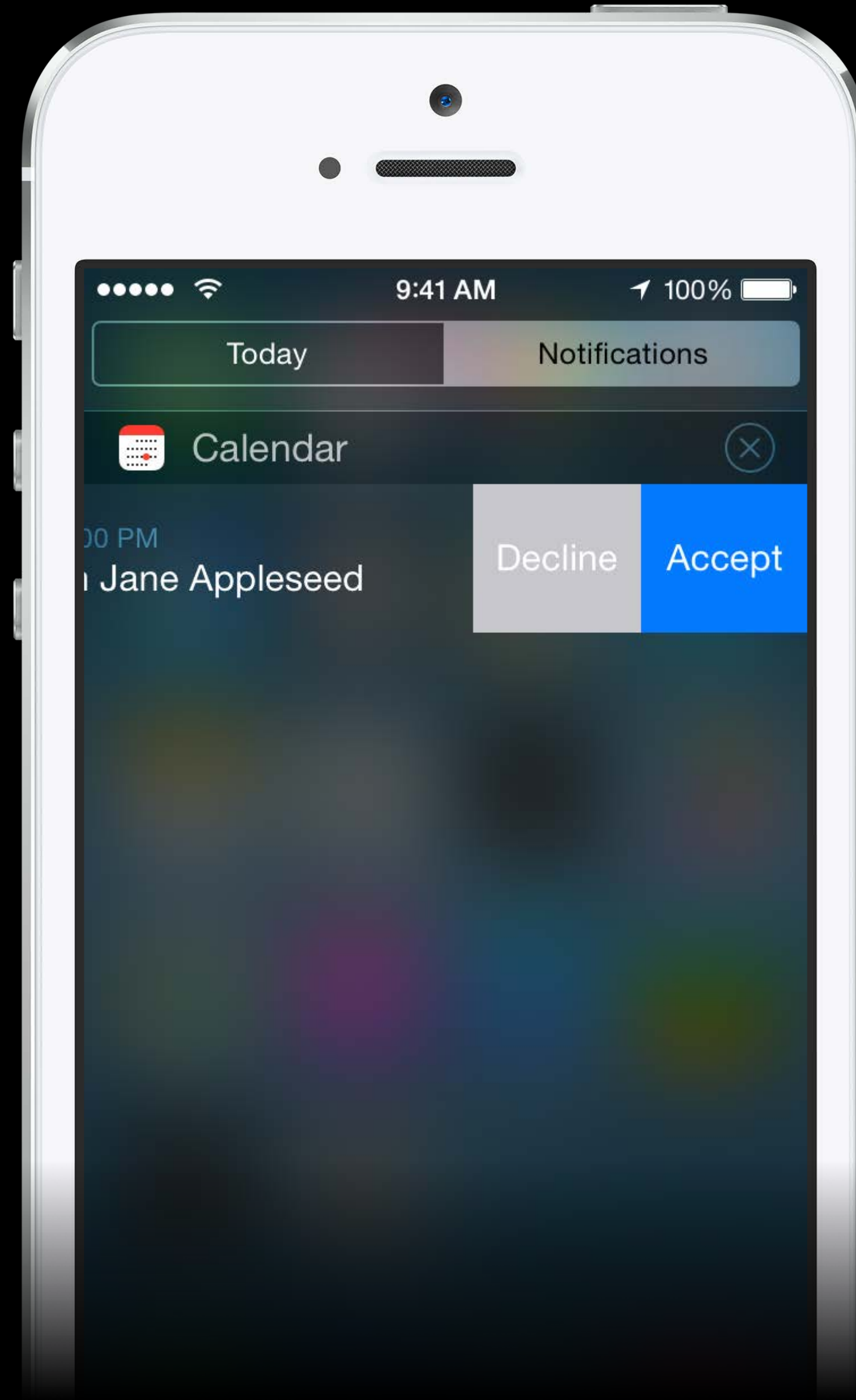
Notification Actions

In iOS 8



Notification Actions

In iOS 8



Notification Actions

In iOS 8



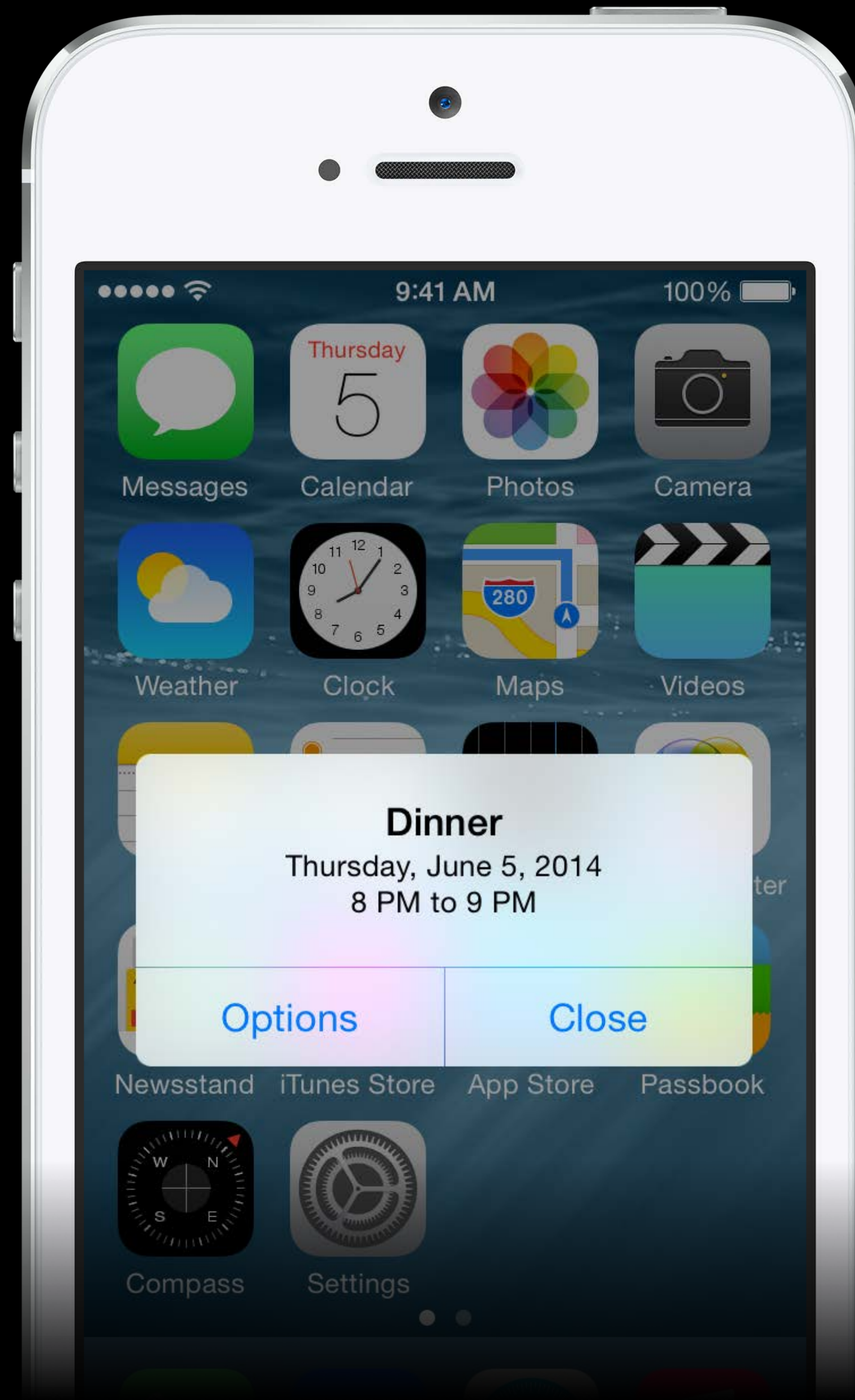
Notification Actions

In iOS 8



Notification Actions

In iOS 8



Notification Actions

In iOS 8



Using Notification Actions

Register Actions

Push/Schedule Notification

Handle Action

Using Notification Actions



Using Notification Actions

Register Actions

Push/Schedule Notification

Handle Action

Using Notification Actions

Register Actions

Push/Schedule Notification

Handle Action

Notification Actions Registration



Actions

```
UIMutableUserNotificationAction *acceptAction =  
[[UIMutableUserNotificationAction alloc] init];  
  
acceptAction.identifier = @"ACCEPT_IDENTIFIER";  
  
acceptAction.title = @"Accept";  
  
// Given seconds, not minutes, to run in the background  
acceptAction.activationMode = UIUserNotificationActivationModeBackground;  
  
acceptAction.destructive = NO;  
  
// If YES requires passcode, but does not unlock the device  
acceptAction.authenticationRequired = NO;
```

Notification Actions Registration

Actions



```
UIMutableUserNotificationAction *acceptAction =  
[[UIMutableUserNotificationAction alloc] init];
```

```
acceptAction.identifier = @"ACCEPT_IDENTIFIER";
```

```
acceptAction.title = @"Accept";
```

```
// Given seconds, not minutes, to run in the background
```

```
acceptAction.activationMode = UIUserNotificationActivationModeBackground;
```

```
acceptAction.destructive = NO;
```

```
// If YES requires passcode, but does not unlock the device
```

```
acceptAction.authenticationRequired = NO;
```

Notification Actions Registration



Actions

```
UIMutableUserNotificationAction *acceptAction =  
[[UIMutableUserNotificationAction alloc] init];
```

```
acceptAction.identifier = @"ACCEPT_IDENTIFIER";
```

```
acceptAction.title = @"Accept";
```

```
// Given seconds, not minutes, to run in the background  
acceptAction.activationMode = UIUserNotificationActivationModeBackground;
```

```
acceptAction.destructive = NO;
```

```
// If YES requires passcode, but does not unlock the device  
acceptAction.authenticationRequired = NO;
```


Notification Actions Registration



Actions

```
UIMutableUserNotificationAction *acceptAction =  
[[UIMutableUserNotificationAction alloc] init];  
  
acceptAction.identifier = @"ACCEPT_IDENTIFIER";  
  
acceptAction.title = @"Accept";  
  
// Given seconds, not minutes, to run in the background  
acceptAction.activationMode = UIUserNotificationActivationModeBackground;  
  
acceptAction.destructive = NO;  
  
// If YES requires passcode, but does not unlock the device  
acceptAction.authenticationRequired = NO;
```

Notification Actions Registration



Actions

```
UIMutableUserNotificationAction *acceptAction =  
[[UIMutableUserNotificationAction alloc] init];
```

```
acceptAction.identifier = @"ACCEPT_IDENTIFIER";
```

```
acceptAction.title = @"Accept";
```

```
// Given seconds, not minutes, to run in the background  
acceptAction.activationMode = UIUserNotificationActivationModeBackground;
```

```
acceptAction.destructive = NO;
```

```
// If YES requires passcode, but does not unlock the device  
acceptAction.authenticationRequired = NO;
```

Notification Actions Registration



Actions

```
UIMutableUserNotificationAction *acceptAction =  
[[UIMutableUserNotificationAction alloc] init];  
  
acceptAction.identifier = @"ACCEPT_IDENTIFIER";  
  
acceptAction.title = @"Accept";  
  
// Given seconds, not minutes, to run in the background  
acceptAction.activationMode = UIUserNotificationActivationModeBackground;  
  
acceptAction.destructive = NO;  
  
// If YES requires passcode, but does not unlock the device  
acceptAction.authenticationRequired = NO;
```

Notification Actions Registration

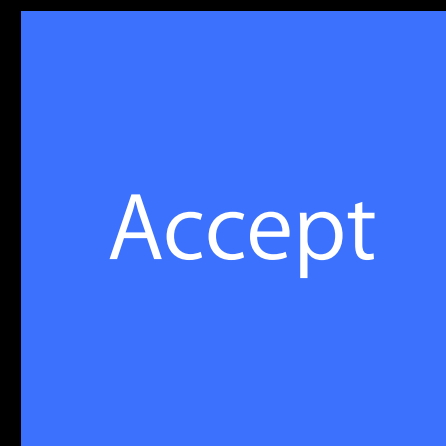


Actions

```
UIMutableUserNotificationAction *acceptAction =  
[[UIMutableUserNotificationAction alloc] init];  
  
acceptAction.identifier = @"ACCEPT_IDENTIFIER";  
  
acceptAction.title = @"Accept";  
  
// Given seconds, not minutes, to run in the background  
acceptAction.activationMode = UIUserNotificationActivationModeBackground;  
  
acceptAction.destructive = NO;  
  
// If YES requires passcode, but does not unlock the device  
acceptAction.authenticationRequired = NO;
```

Notification Actions Registration

Accept invite



Action Property

Value

title

@“Accept”

activationMode

UIUserNotificationActivationModeBackground

destructive

NO

authenticationRequired

NO

Notification Actions Registration

Trash mail



Action Property

Value

title

@“Trash”

activationMode

UIUserNotificationActivationModeBackground

destructive

YES

authenticationRequired

YES

Notification Actions Registration

Reply to message



Action Property

Value

title

@“Reply”

activationMode

UIUserNotificationActivationModeForeground

destructive

NO

authenticationRequired

YES*

Notification Actions Registration

Categories

Category

Actions

Invite

Accept, Maybe, Decline

New mail

Mark as Read, Trash

Tagged

Like, Comment, Untag

Notification Actions Registration

Categories



```
UIMutableUserNotificationCategory *inviteCategory =  
[[UIMutableUserNotificationCategory alloc] init];  
  
inviteCategory.identifier = @"INVITE_CATEGORY";  
  
[inviteCategory setActions:@[acceptAction, maybeAction, declineAction]  
forContext:UIUserNotificationActionContextDefault];
```

Notification Actions Registration

Categories



```
UIMutableUserNotificationCategory *inviteCategory =  
[[UIMutableUserNotificationCategory alloc] init];
```

```
inviteCategory.identifier = @"INVITE_CATEGORY";
```

```
[inviteCategory setActions:@[acceptAction, maybeAction, declineAction]  
forContext:UIUserNotificationActionContextDefault];
```

Notification Actions Registration

Categories



```
UIMutableUserNotificationCategory *inviteCategory =  
[[UIMutableUserNotificationCategory alloc] init];
```

```
inviteCategory.identifier = @"INVITE_CATEGORY";
```

```
[inviteCategory setActions:@[acceptAction, maybeAction, declineAction]  
forContext:UIUserNotificationActionContextDefault];
```

Notification Actions Registration

Categories

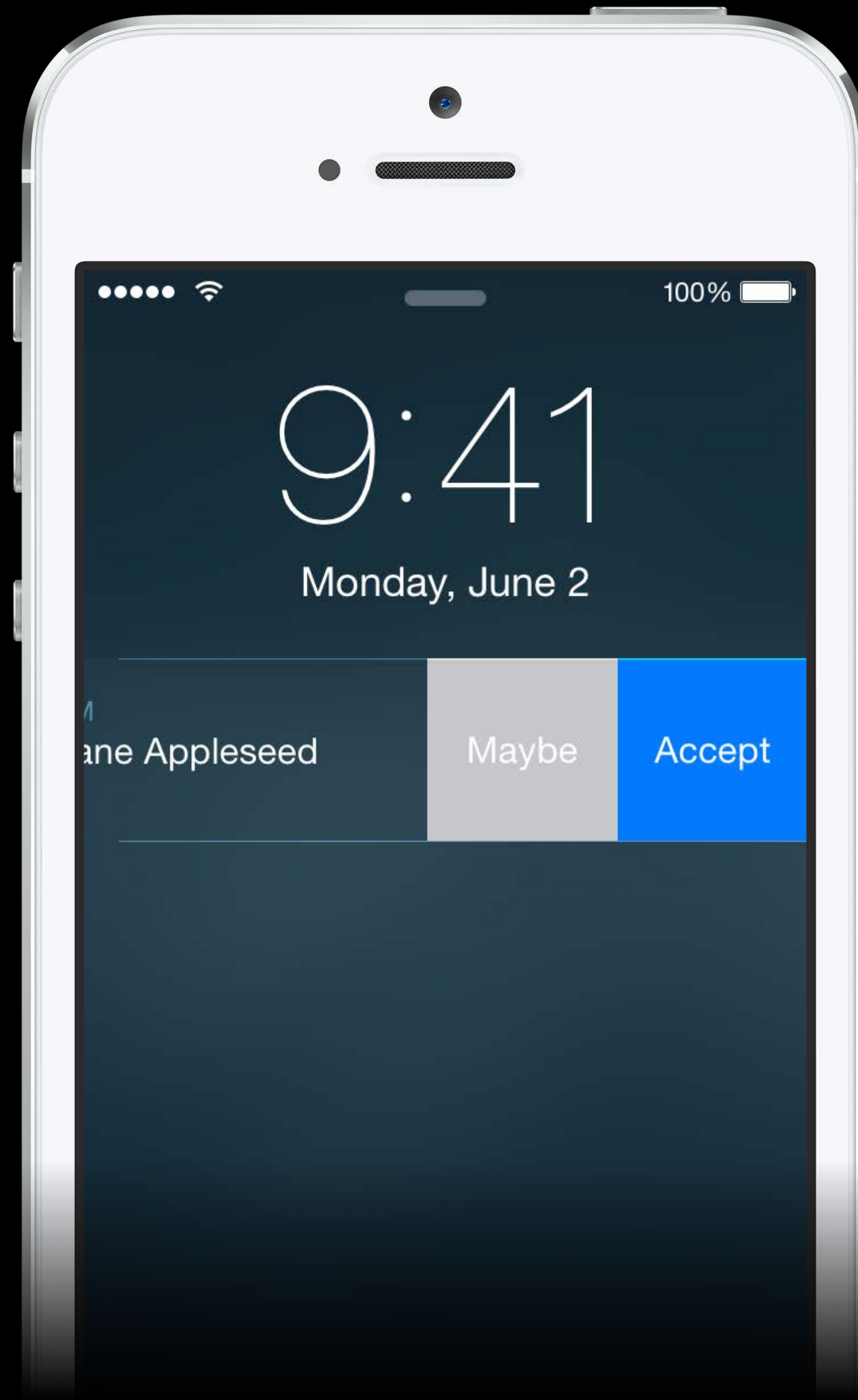


```
UIMutableUserNotificationCategory *inviteCategory =  
[[UIMutableUserNotificationCategory alloc] init];
```

```
inviteCategory.identifier = @"INVITE_CATEGORY";
```

```
[inviteCategory setActions:@[acceptAction, maybeAction, declineAction]  
forContext:UIUserNotificationActionContextDefault];
```

Notification Actions Registration



Notification Actions Registration

Categories



```
UIMutableUserNotificationCategory *inviteCategory =  
[[UIMutableUserNotificationCategory alloc] init];  
  
inviteCategory.identifier = @"INVITE_CATEGORY";  
  
[inviteCategory setActions:@[acceptAction, maybeAction, declineAction]  
forContext:UIUserNotificationActionContextDefault];  
  
[inviteCategory setActions:@[acceptAction, declineAction]  
forContext:UIUserNotificationActionContextMinimal];
```

Notification Actions Registration

Categories



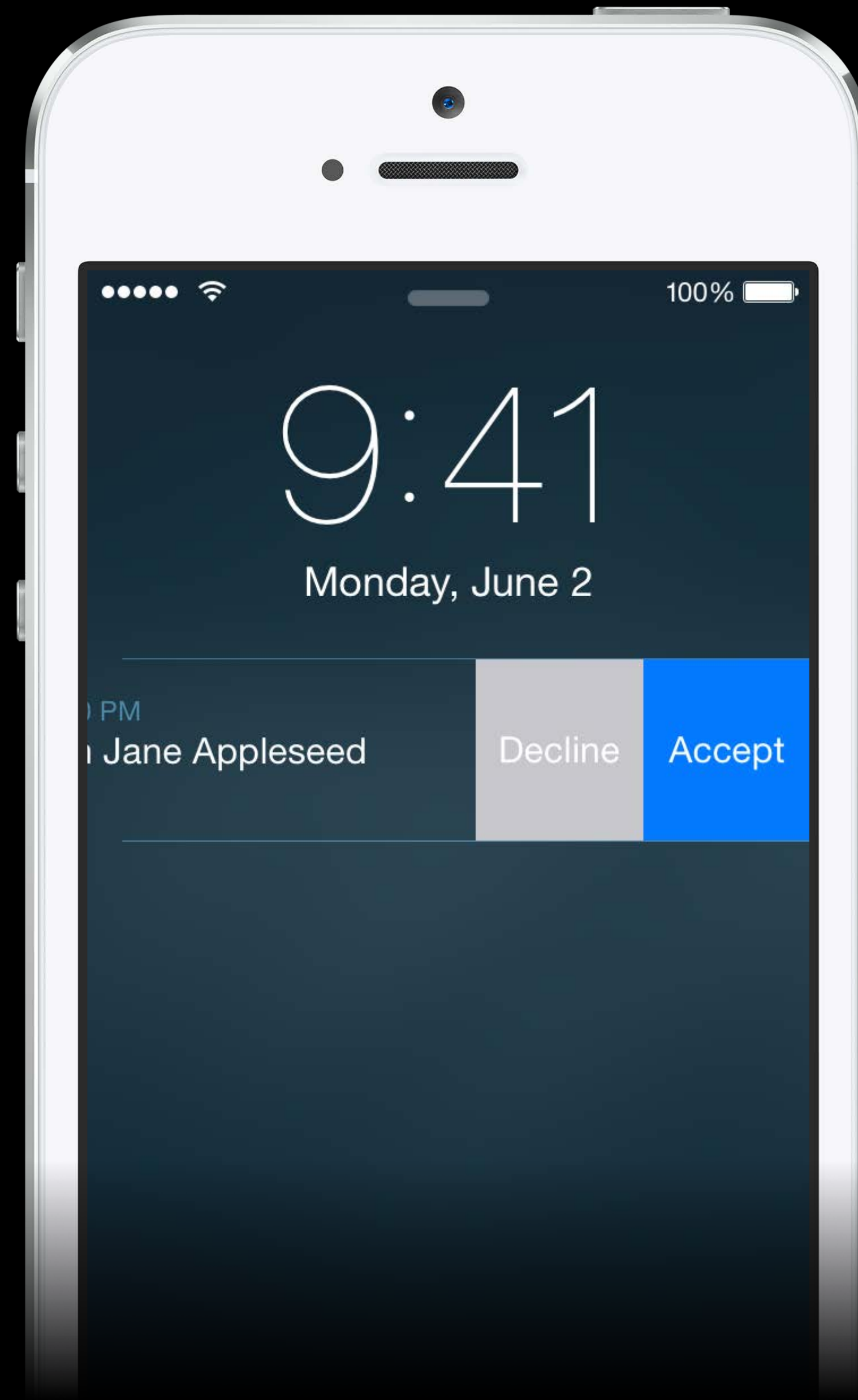
```
UIMutableUserNotificationCategory *inviteCategory =  
[[UIMutableUserNotificationCategory alloc] init];
```

```
inviteCategory.identifier = @"INVITE_CATEGORY";
```

```
[inviteCategory setActions:@[acceptAction, maybeAction, declineAction]  
forContext:UIUserNotificationActionContextDefault];
```

```
[inviteCategory setActions:@[acceptAction, declineAction]  
forContext:UIUserNotificationActionContextMinimal];
```

Notification Actions Registration



Notification Actions Registration



Notification Actions Registration

Settings



```
NSSet *categories = [NSSet setWithObjects:inviteCategory, alarmCategory, ...

UIUserNotificationSettings *settings =
    [UIUserNotificationSettings settingsForTypes:types categories:categories];

[[UIApplication sharedApplication]
registerUserNotificationSettings:settings];
```

Notification Actions Registration

Settings



```
NSSet *categories = [NSSet setWithObjects:inviteCategory, alarmCategory, ...
```

```
UIUserNotificationSettings *settings =  
    [UIUserNotificationSettings settingsForTypes:types categories:categories];  
  
[[UIApplication sharedApplication]  
registerUserNotificationSettings:settings];
```

Notification Actions Registration

Settings



```
NSSet *categories = [NSSet setWithObjects:inviteCategory, alarmCategory, ...
```

```
UIUserNotificationSettings *settings =  
    [UIUserNotificationSettings settingsForTypes:types categories:categories];
```

```
[[UIApplication sharedApplication]  
registerUserNotificationSettings:settings];
```

Notification Actions Registration

Settings



```
NSSet *categories = [NSSet setWithObjects:inviteCategory, alarmCategory, ...

UIUserNotificationSettings *settings =
    [UIUserNotificationSettings settingsForTypes:types categories:categories];

[[UIApplication sharedApplication]
registerUserNotificationSettings:settings];
```

Using Notification Actions

Register Actions

Push/Schedule Notification

Handle Action

Using Notification Actions

Register Actions

Push/Schedule Notification

Handle Action

Remote Notifications

```
{  
  "aps" : {  
    "alert" : "You're invited!",  
    "category" : "INVITE_CATEGORY",  
  }  
}
```


Remote Notifications

```
{  
  "aps" : {  
    "alert" : "You're invited!",  
    "category" : "INVITE_CATEGORY",  
  }  
}
```

Remote Notifications

```
{  
  "aps" : {  
    "alert" : "You're invited!",  
    "category" : "INVITE_CATEGORY",  
  }  
}
```

2KB Max Size

Local Notifications

```
UILocalNotification *notification = [[UILocalNotification alloc] init];  
...  
notification.category = @"INVITE_CATEGORY";  
  
[[UIApplication sharedApplication] scheduleLocalNotification:notification];
```

Local Notifications

```
UILocalNotification *notification = [[UILocalNotification alloc] init];  
...
```

```
notification.category = @"INVITE_CATEGORY";
```

```
[[UIApplication sharedApplication] scheduleLocalNotification:notification];
```

Local Notifications

```
UILocalNotification *notification = [[UILocalNotification alloc] init];  
...
```

```
notification.category = @"INVITE_CATEGORY";
```

```
[[UIApplication sharedApplication] scheduleLocalNotification:notification];
```

Local Notifications

```
UILocalNotification *notification = [[UILocalNotification alloc] init];
```

```
...
```

```
notification.category = @"INVITE_CATEGORY";
```

```
[[UIApplication sharedApplication] scheduleLocalNotification:notification];
```

Using Notification Actions

Register Actions

Push/Schedule Notification

Handle Action

Using Notification Actions

Register Actions

Push/Schedule Notification

Handle Action

Handling Notification Actions



App

Handling Notification Actions



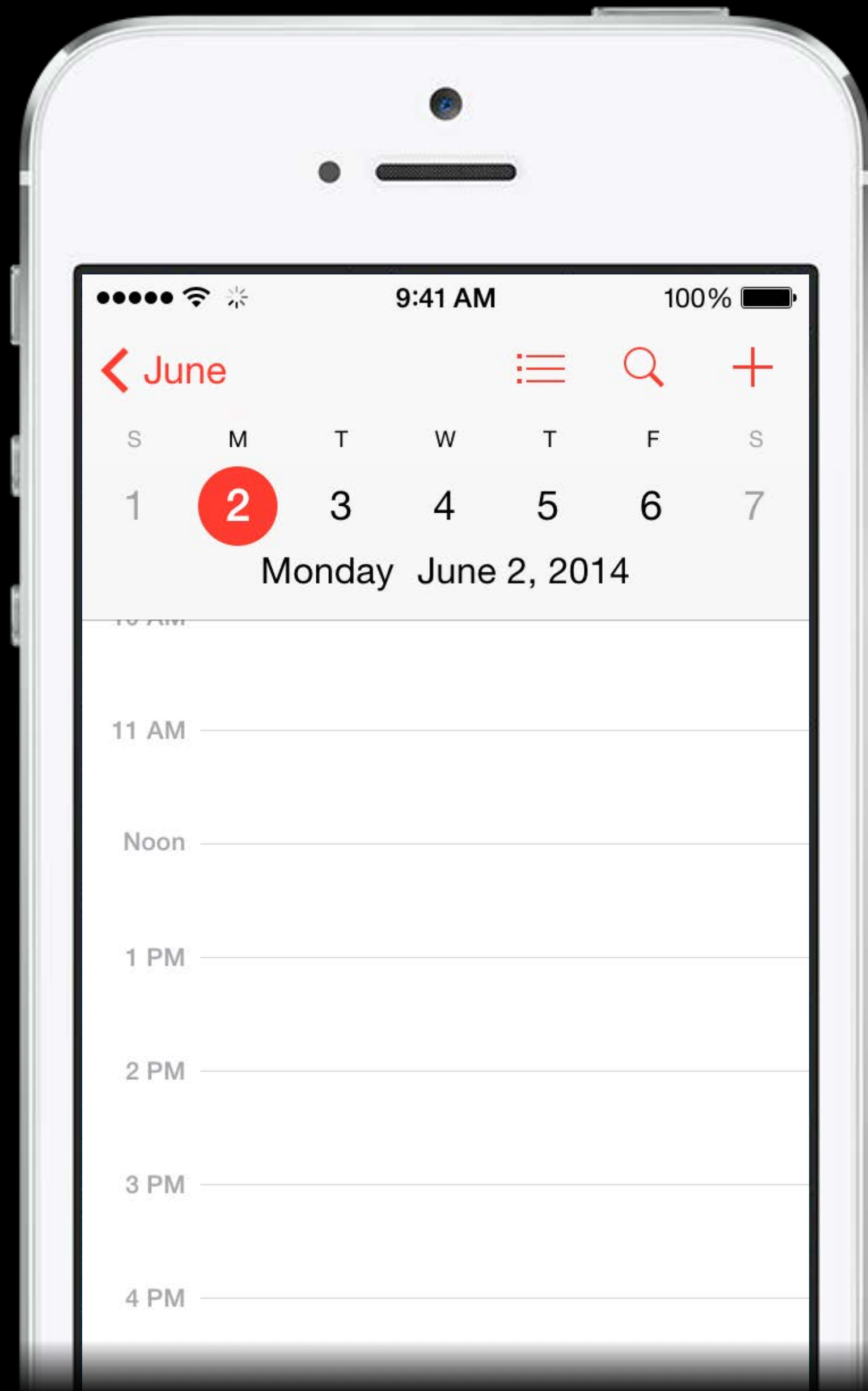
App

UIApplicationDelegate

```
application:didFinishLaunchingWithOptions:
```

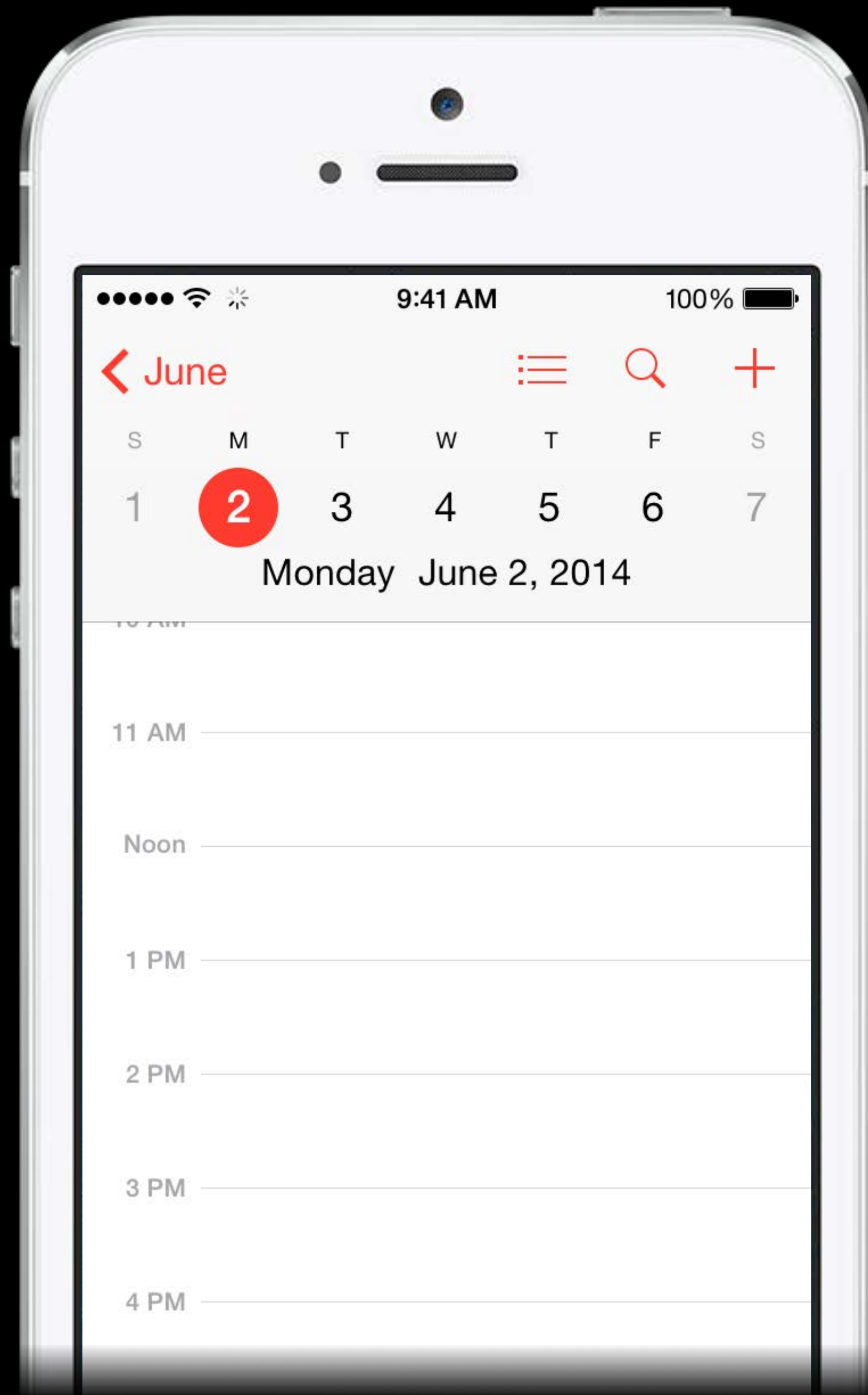
```
application:didReceiveRemoteNotification:  
fetchCompletionHandler:
```

Handling Notification Actions



App

Handling Notification Actions



App

UIApplicationDelegate

```
application:didReceiveLocalNotification:
```

```
application:didReceiveRemoteNotification:
```

```
application:didReceiveRemoteNotification:  
fetchCompletionHandler:
```

Handling Notification Actions



App

Handling Notification Actions

NEW



App

UIApplicationDelegate

```
application:handleActionWithIdentifier:  
forRemoteNotification:completionHandler:
```

```
application:handleActionWithIdentifier:  
forLocalNotification:completionHandler:
```

Handling Notification Actions



UIApplicationDelegate

```
- (void)application:(UIApplication *)application
    handleActionWithIdentifier:(NSString *)identifier
    forRemoteNotification:(NSDictionary *)notification
    completionHandler:(void (^)(void))completionHandler {

    if ([identifier isEqualToString:@"ACCEPT_IDENTIFIER"]) {
        [self handleAcceptActionWithNotification:notification];
    }

    // Must be called when finished
    completionHandler();
}
```

Handling Notification Actions



UIApplicationDelegate

```
- (void)application:(UIApplication *)application
    handleActionWithIdentifier:(NSString *)identifier
        forRemoteNotification:(NSDictionary *)notification
        completionHandler:(void (^)(void))completionHandler {

    if ([identifier isEqualToString:@"ACCEPT_IDENTIFIER"]) {
        [self handleAcceptActionWithNotification:notification];
    }

    // Must be called when finished
    completionHandler();
}
```


Handling Notification Actions



UIApplicationDelegate

```
- (void)application:(UIApplication *)application
    handleActionWithIdentifier:(NSString *)identifier
        forRemoteNotification:(NSDictionary *)notification
        completionHandler:(void (^)(void))completionHandler {
    if ([identifier isEqualToString:@"ACCEPT_IDENTIFIER"]) {
        [self handleAcceptActionWithNotification:notification];
    }

    // Must be called when finished
    completionHandler();
}
```

Handling Notification Actions



UIApplicationDelegate

```
- (void)application:(UIApplication *)application
    handleActionWithIdentifier:(NSString *)identifier
        forRemoteNotification:(NSDictionary *)notification
        completionHandler:(void (^)(void))completionHandler {

    if ([identifier isEqualToString:@"ACCEPT_IDENTIFIER"]) {
        [self handleAcceptActionWithNotification:notification];
    }

    // Must be called when finished
    completionHandler();
}
```

Handling Notification Actions



UIApplicationDelegate

```
- (void)application:(UIApplication *)application
    handleActionWithIdentifier:(NSString *)identifier
        forLocalNotification:(UILocalNotification *)notification
        completionHandler:(void (^)(void))completionHandler {

    if ([identifier isEqualToString:@"ACCEPT_IDENTIFIER"]) {
        [self handleAcceptActionWithNotification:notification];
    }

    // Must be called when finished
    completionHandler();
}
```

Handling Notification Actions



UIApplicationDelegate

```
- (void)application:(UIApplication *)application
    handleActionWithIdentifier:(NSString *)identifier
        forLocalNotification:(UILocalNotification *)notification
        completionHandler:(void (^)(void))completionHandler {

    if ([identifier isEqualToString:@"ACCEPT_IDENTIFIER"]) {
        [self handleAcceptActionWithNotification:notification];
    }

    // Must be called when finished
    completionHandler();
}
```

Handling Notification Actions



UIApplicationDelegate

```
- (void)application:(UIApplication *)application
    handleActionWithIdentifier:(NSString *)identifier
        forLocalNotification:(UILocalNotification *)notification
            completionHandler:(void (^)(void))completionHandler {

    if ([identifier isEqualToString:@"ACCEPT_IDENTIFIER"]) {
        [self handleAcceptActionWithNotification:notification];
    }

    // Must be called when finished
    completionHandler();
}
```

Handling Notification Actions



UIApplicationDelegate

```
- (void)application:(UIApplication *)application
    handleActionWithIdentifier:(NSString *)identifier
        forLocalNotification:(UILocalNotification *)notification
        completionHandler:(void (^)(void))completionHandler {

    if ([identifier isEqualToString:@"ACCEPT_IDENTIFIER"]) {
        [self handleAcceptActionWithNotification:notification];
    }

    // Must be called when finished
    completionHandler();
}
```

Using Notification Actions

Register Actions

Push/Schedule Notification

Handle Action

Using Notification Actions

Register Actions

UIUserNotificationAction

UIUserNotificationCategory

UIUserNotificationSettings

Push/Schedule Notification

Handle Action

Using Notification Actions

Register Actions

Push/Schedule Notification

```
aps {  
  alert: {...}  
  category: "INVITE"  
}
```

```
notification.category = @"INVITE";
```

Handle Action

Using Notification Actions

Register Actions

Push/Schedule Notification

Handle Action

UIApplicationDelegate

```
application:handleActionWithIdentifier:  
forRemoteNotification:completionHandler:
```

```
application:handleActionWithIdentifier:  
forLocalNotification:completionHandler:
```

Remote Notifications

Remote Notifications



APNs

Remote Notifications



APNs



```
aps {  
  alert: {...}  
}
```

Remote Notifications



APNs

Remote Notifications



App



APNs

Remote Notifications



App



APNs



```
aps {  
  content-available: 1  
}
```


Remote Notifications



Content Server



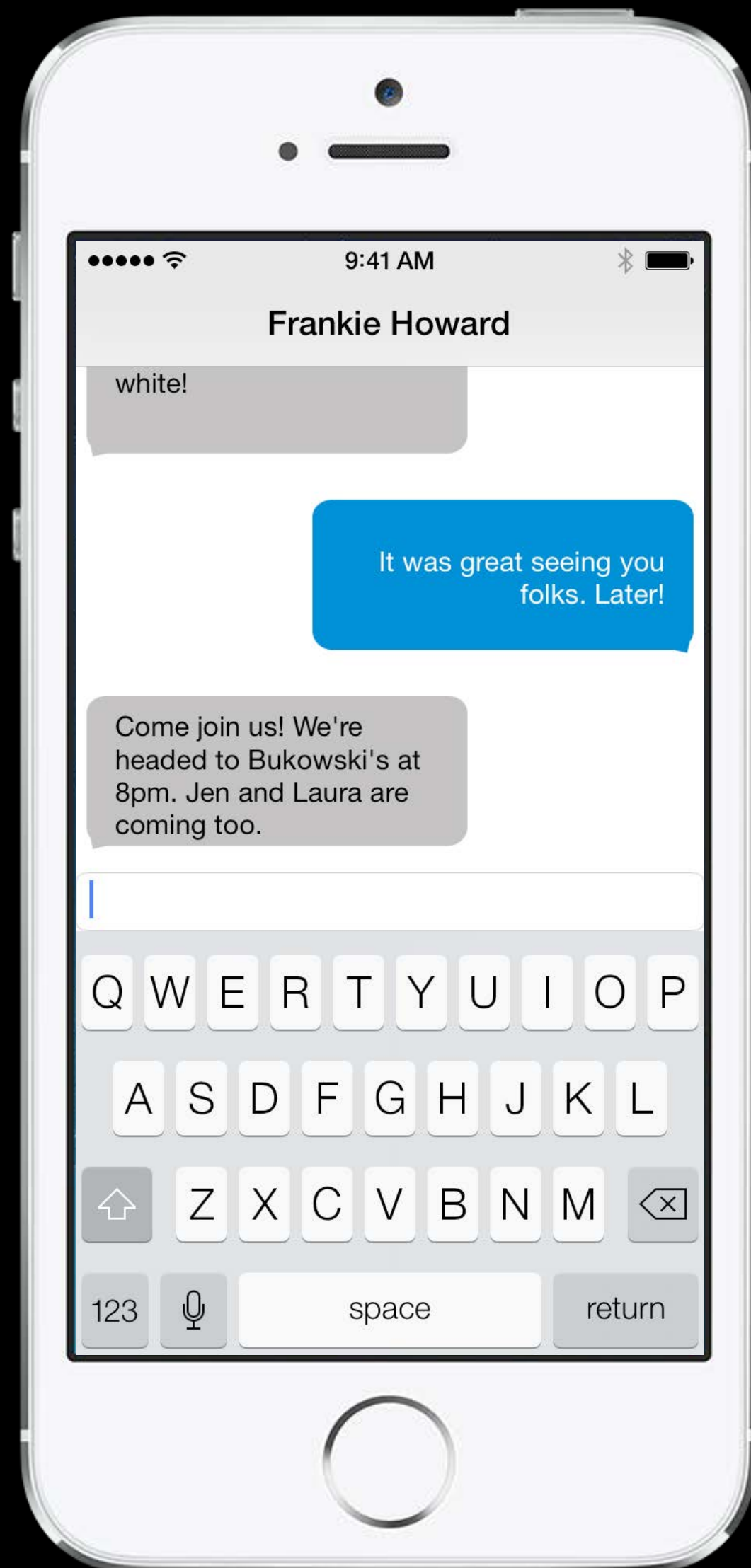
App



APNs

```
aps {  
  content-available: 1  
}
```

Remote Notifications



Remote Notifications

User

- Requires call to `registerUserNotificationSettings:`

Silent

- Info.plist `UIBackgroundModes` array contains `remote-notification`

Can use both

Remote Notifications

Must register before receiving pushes

Enabled by default

Can be disabled via Settings

Remote Notifications

Registering



```
UIApplication *myApp = [UIApplication sharedApplication];
```

```
[myApp registerForRemoteNotificationTypes:someTypes];
```

Remote Notifications

Registering



```
UIApplication *myApp = [UIApplication sharedApplication];
```

```
[myApp registerForRemoteNotificationTypes:someTypes];
```

Remote Notifications

Registering



```
UIApplication *myApp = [UIApplication sharedApplication];
```

```
[myApp registerForRemoteNotificationTypes:someTypes];
```



Remote Notifications

Registering



```
UIApplication *myApp = [UIApplication sharedApplication];
```

```
[myApp registerForRemoteNotificationTypes:someTypes];
```

```
[myApp registerForRemoteNotifications];
```


Remote Notifications

Registering



```
UIApplication *myApp = [UIApplication sharedApplication];
```

```
[myApp registerForRemoteNotificationTypes:someTypes];
```

```
[myApp registerForRemoteNotifications];
```



Remote Notifications

Registering



```
UIApplication *myApp = [UIApplication sharedApplication];
```

```
[myApp registerForRemoteNotifications];
```

```
[myApp registerUserNotificationSettings:mySettings];
```

Remote Notifications

UIApplicationDelegate success callback

```
- (void)application:(UIApplication *)application
  didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {

    MyAPNSServer *myServer = // ...
    [myServer registerPushToken:deviceToken];

}
```

Remote Notifications

UIApplicationDelegate success callback

```
- (void)application:(UIApplication *)application
    didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {

    MyAPNSServer *myServer = // ...
    [myServer registerPushToken:deviceToken];

}
```

Remote Notifications

UIApplicationDelegate success callback

```
- (void)application:(UIApplication *)application
    didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {

    MyAPNSServer *myServer = // ...
    [myServer registerPushToken:deviceToken];

}
```

Remote Notifications

UIApplicationDelegate failure callback

```
- (void)application:(UIApplication *)application
    didFailToRegisterForRemoteNotificationsWithError:(NSError *)error {

    [self handleRegistrationError:error];

}
```

Remote Notifications

UIApplicationDelegate failure callback

```
- (void)application:(UIApplication *)application  
    didFailToRegisterForRemoteNotificationsWithError:(NSError *)error {  
  
    [self handleRegistrationError:error];  
  
}
```

Remote Notifications

UIApplicationDelegate failure callback

```
- (void)application:(UIApplication *)application  
    didFailToRegisterForRemoteNotificationsWithError:(NSError *)error {  
  
    [self handleRegistrationError:error];  
  
}
```


Remote Notifications

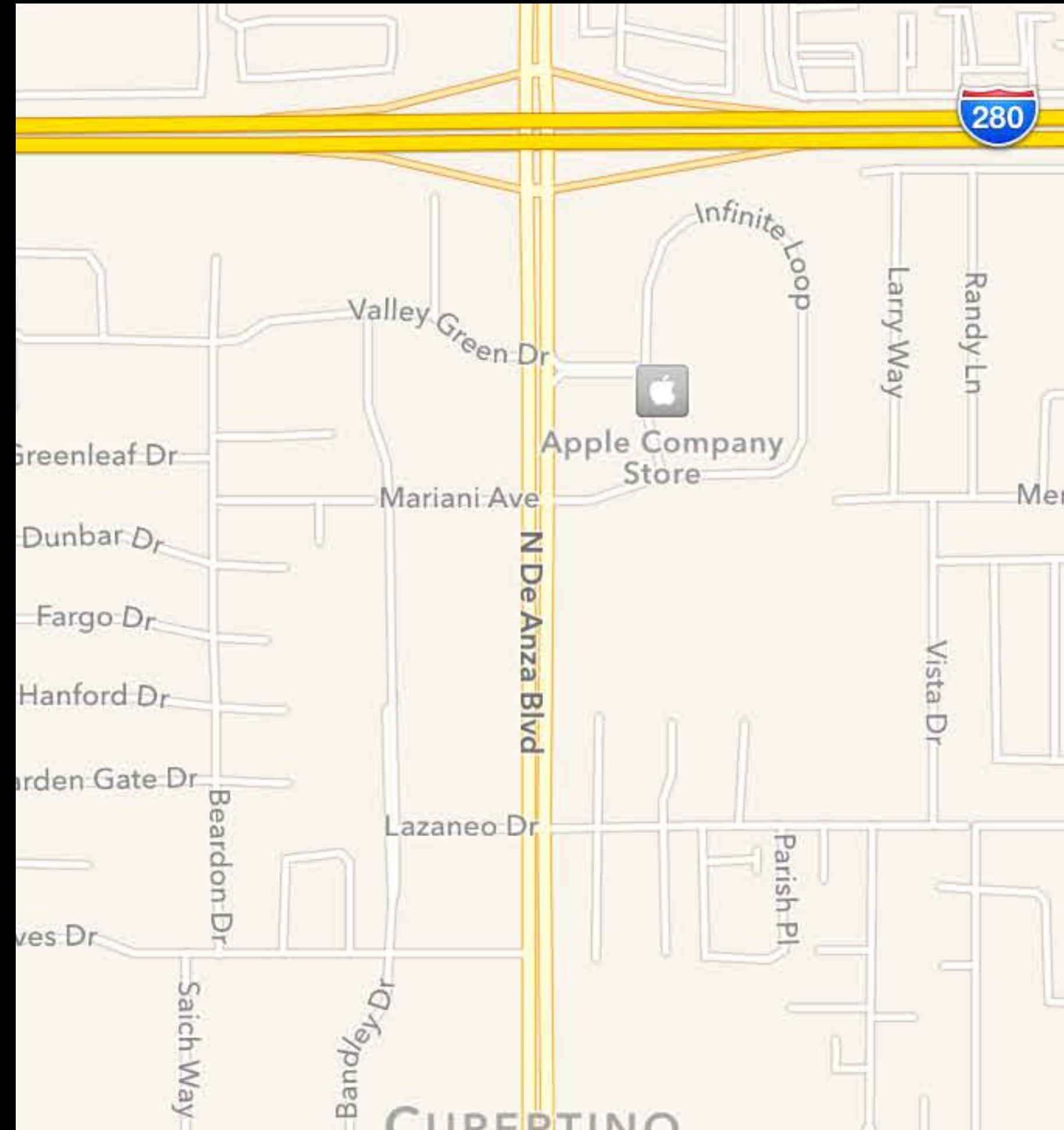
Wrap up

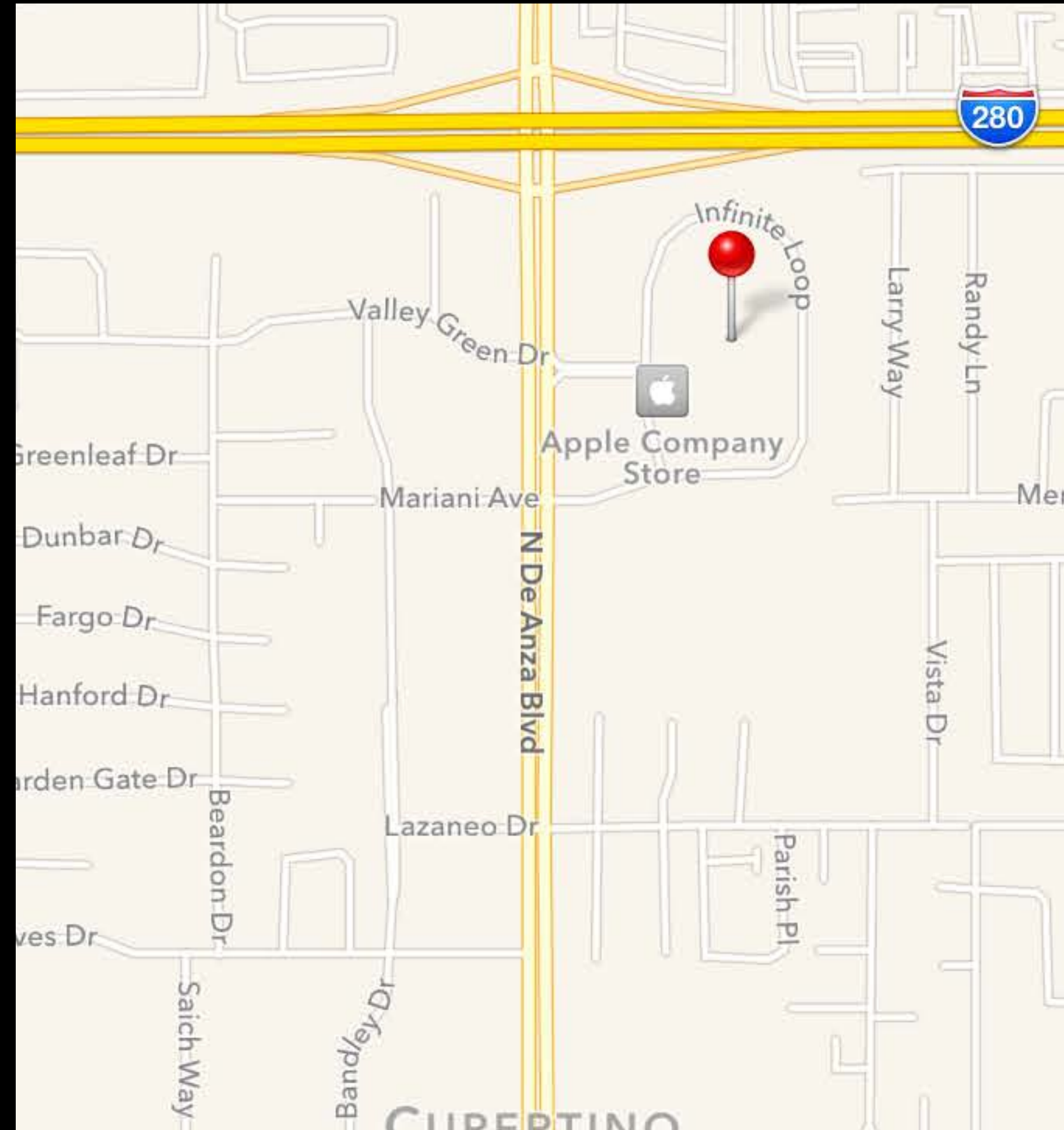
Must register before using

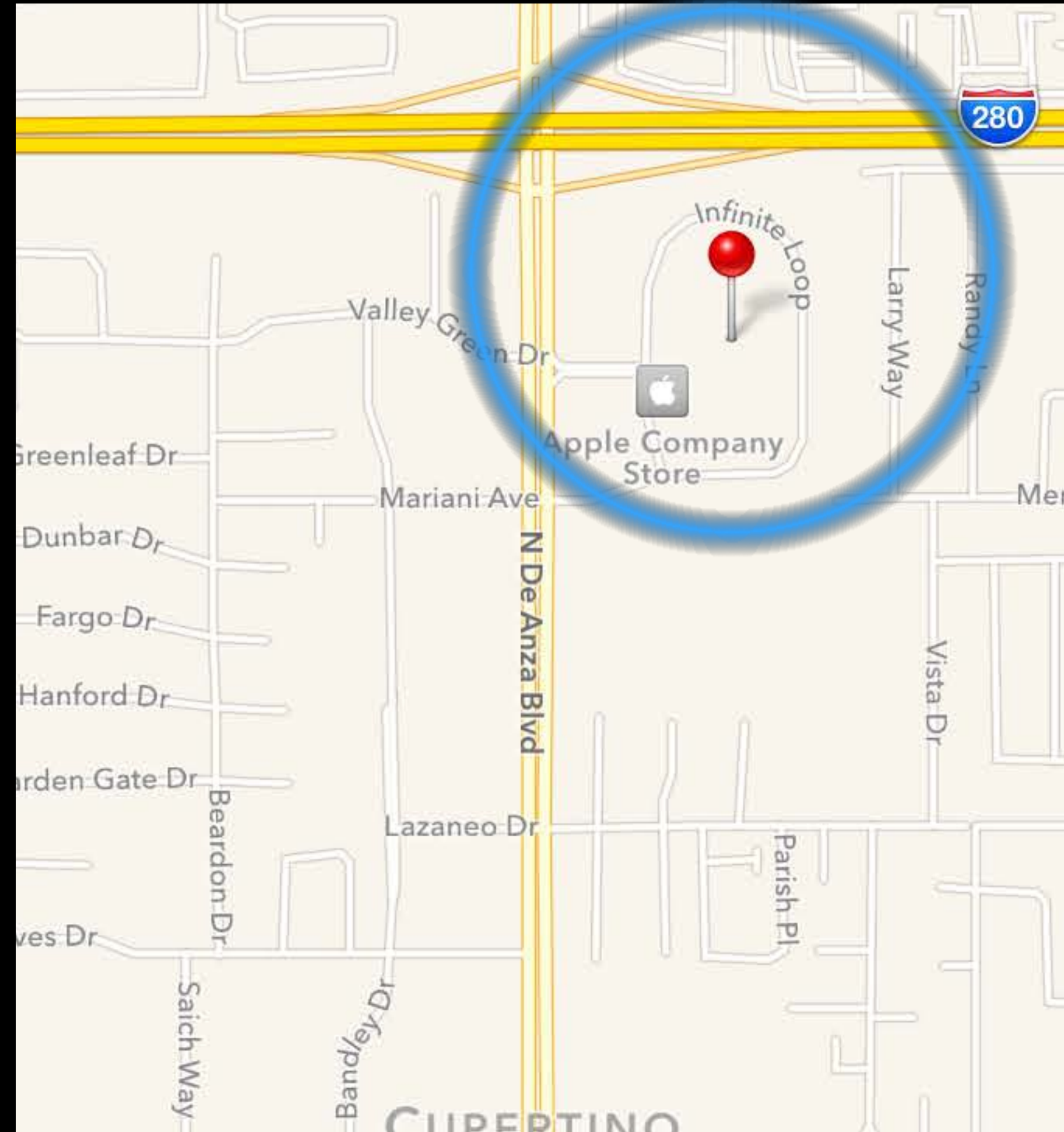
Enabled by default

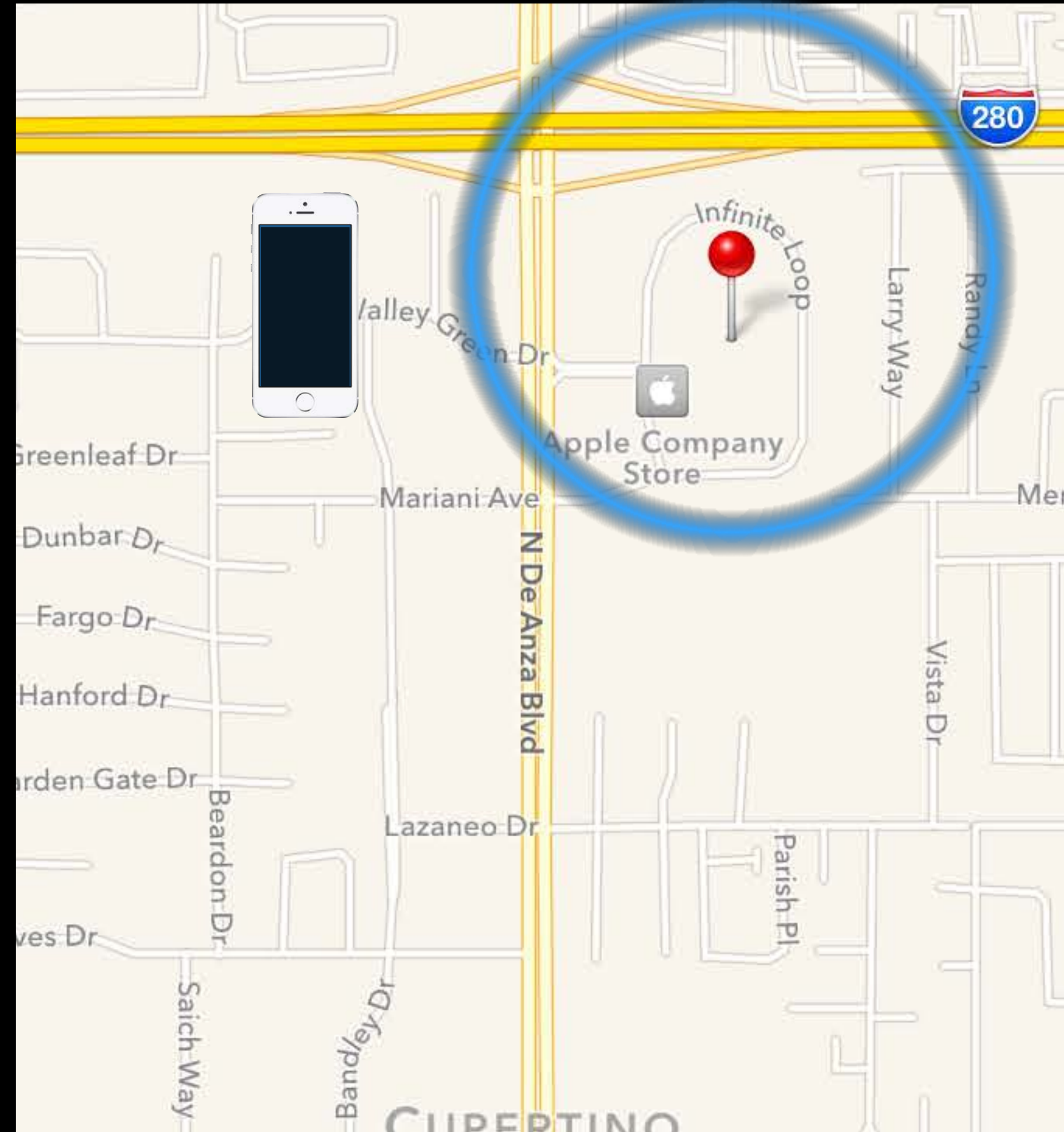
Register API change (`registerForRemoteNotifications`)

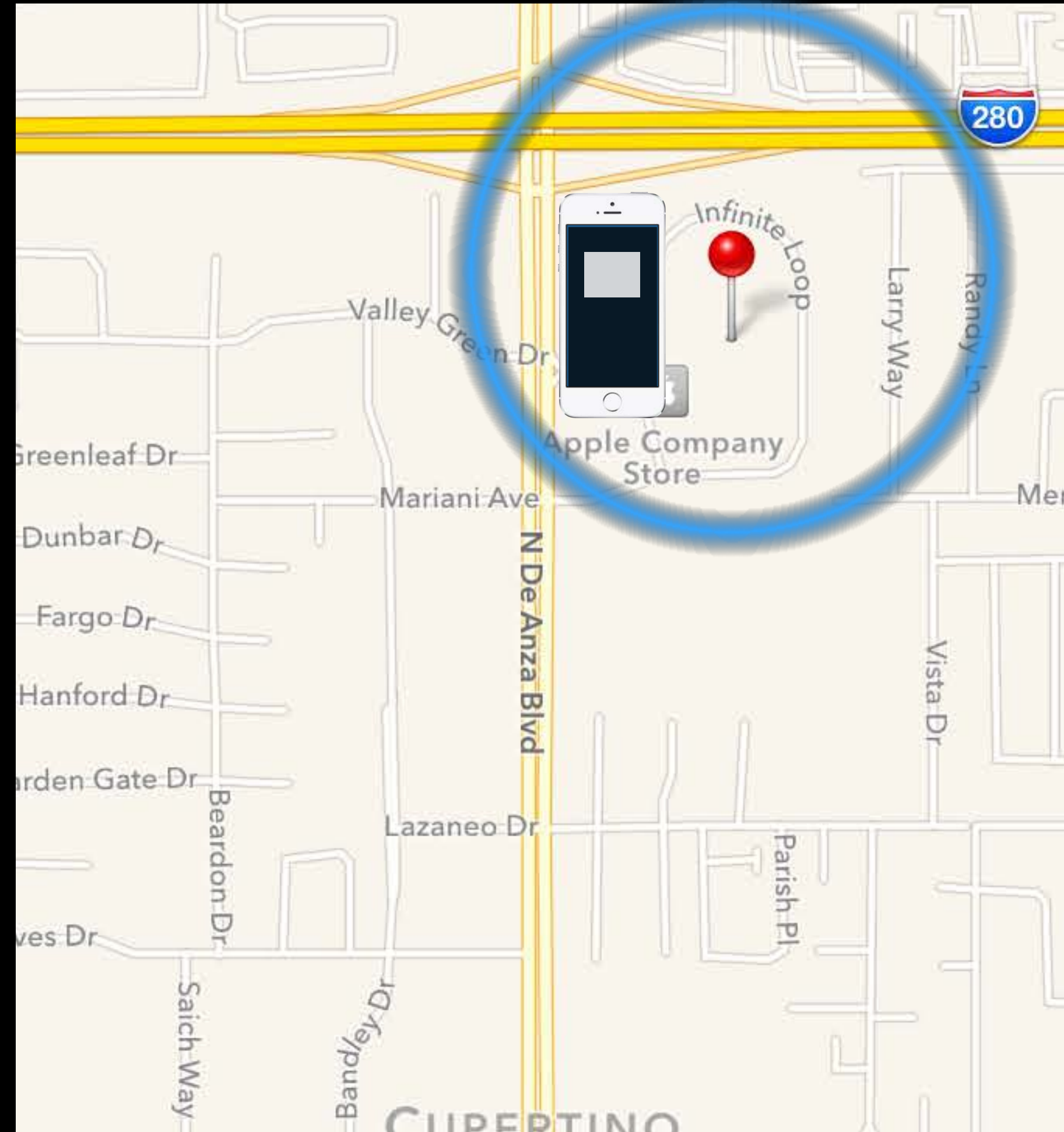
Location Notifications











Location Notifications

NEW

Uses `UILocalNotification`

Fire when user enters or exits a region

Fire-Once or Fire-Always

Requires Core Location registration



Location Notifications

Core Location registration



```
CLLocationManager *locMan = [[CLLocationManager alloc] init];
```

```
locMan.delegate = self;
```

```
// request authorization to track the user's location  
[locMan requestWhenInUseAuthorization];
```

Location Notifications

Core Location registration



```
CLLocationManager *locMan = [[CLLocationManager alloc] init];
```

```
locMan.delegate = self;
```

```
// request authorization to track the user's location  
[locMan requestWhenInUseAuthorization];
```

Location Notifications

Core Location registration



```
CLLocationManager *locMan = [[CLLocationManager alloc] init];
```

```
locMan.delegate = self;
```

```
// request authorization to track the user's location  
[locMan requestWhenInUseAuthorization];
```

Location Notifications

Core Location registration



```
CLLocationManager *locMan = [[CLLocationManager alloc] init];
```

```
locMan.delegate = self;
```

```
// request authorization to track the user's location  
[locMan requestWhenInUseAuthorization];
```

Location Notifications

Core Location registration



```
CLLocationManager *locMan = [[CLLocationManager alloc] init];
```

```
locMan.delegate = self;
```

```
// request authorization to track the user's location  
[locMan requestWhenInUseAuthorization];
```

**Allow “My App” to Access
Your Location While You Use
the App?**

Enabling location tracking allows
friends to see where you are.

Don't Allow

Allow

Location Notifications

Core Location registration



Key	Type	Type
▼ Information Property List	Dictionary	(17 items)
NSLocationWhenInUseUsageDescription	String	Enabling location tracking allows friends to see where you are.
Localization native development region	String	en
Bundle display name	String	\${PRODUCT_NAME}
Executable file	String	\${EXECUTABLE_NAME}

Location Notifications

Core Location registration



Key	Type	Type
▼ Information Property List	Dictionary	(17 items)
NSLocationWhenInUseUsageDescription	String	Enabling location tracking allows friends to see where you are.
Localization native development region	String	en
Bundle display name	String	\${PRODUCT_NAME}
Executable file	String	\${EXECUTABLE_NAME}

Location Notifications

Core Location registration callbacks



```
- (void) locationManager: (CLLocationManager *) manager
  didChangeAuthorizationStatus: (CAuthorizationStatus) status {

    // check status to see if we're authorized
    BOOL canUseLocationNotifications =
        (status == kCLAuthorizationStatusAuthorizedWhenInUse);

    if (canUseLocationNotifications) {
        [self startShowingLocationNotifications];
    }
}
```


Location Notifications

Core Location registration callbacks



```
- (void) locationManager: (CLLocationManager *) manager
  didChangeAuthorizationStatus: (CLAuthorizationStatus) status {

    // check status to see if we're authorized
    BOOL canUseLocationNotifications =
        (status == kCLAuthorizationStatusAuthorizedWhenInUse);

    if (canUseLocationNotifications) {
        [self startShowingLocationNotifications];
    }

}
```

Location Notifications

Core Location registration callbacks



```
- (void) locationManager: (CLLocationManager *) manager
  didChangeAuthorizationStatus: (CLAuthorizationStatus) status {

    // check status to see if we're authorized
    BOOL canUseLocationNotifications =
      (status == kCLAuthorizationStatusAuthorizedWhenInUse);

    if (canUseLocationNotifications) {
      [self startShowingLocationNotifications];
    }
  }
}
```

Location Notifications

Core Location registration callbacks



```
- (void)locationManager:(CLLocationManager *)manager
  didChangeAuthorizationStatus:(CAuthorizationStatus)status {

    // check status to see if we're authorized
    BOOL canUseLocationNotifications =
        (status == kCLAuthorizationStatusAuthorizedWhenInUse);

    if (canUseLocationNotifications) {
        [self startShowingLocationNotifications];
    }
}
```

Location Notifications

Notification scheduling

```
- (void)startShowingNotifications {
    UILocalNotification *locNotification = [[UILocalNotification alloc]
                                           init];
    locNotification.alertBody = @"You have arrived!";
    locNotification.regionTriggersOnce = YES;

    locNotification.region = [[CLCircularRegion alloc]
                              initWithCenter:LOC_COORDINATE
                              radius:LOC_RADIUS
                              identifier:LOC_IDENTIFIER];

    [[UIApplication sharedApplication]
     scheduleLocalNotification:locNotification];
}
```

Location Notifications

Notification scheduling

```
- (void)startShowingNotifications {  
    UILocalNotification *locNotification = [[UILocalNotification alloc]  
                                           init];  
    locNotification.alertBody = @"You have arrived!";  
    locNotification.regionTriggersOnce = YES;  
  
    locNotification.region = [[CLCircularRegion alloc]  
                             initWithCenter:LOC_COORDINATE  
                             radius:LOC_RADIUS  
                             identifier:LOC_IDENTIFIER];  
  
    [[UIApplication sharedApplication]  
     scheduleLocalNotification:locNotification];  
}
```

Location Notifications

Notification scheduling

```
- (void)startShowingNotifications {
    UILocalNotification *locNotification = [[UILocalNotification alloc]
                                           init];
    locNotification.alertBody = @"You have arrived!";
    locNotification.regionTriggersOnce = YES;

    locNotification.region = [[CLCircularRegion alloc]
                              initWithCenter:LOC_COORDINATE
                              radius:LOC_RADIUS
                              identifier:LOC_IDENTIFIER];

    [[UIApplication sharedApplication]
     scheduleLocalNotification:locNotification];
}
```

Location Notifications

Notification scheduling

```
- (void)startShowingNotifications {
    UILocalNotification *locNotification = [[UILocalNotification alloc]
                                           init];
    locNotification.alertBody = @"You have arrived!";
    locNotification.regionTriggersOnce = YES;

    locNotification.region = [[CLCircularRegion alloc]
                              initWithCenter:LOC_COORDINATE
                              radius:LOC_RADIUS
                              identifier:LOC_IDENTIFIER];

    [[UIApplication sharedApplication]
     scheduleLocalNotification:locNotification];
}
```

Location Notifications

Notification scheduling

```
- (void)startShowingNotifications {  
    UILocalNotification *locNotification = [[UILocalNotification alloc]  
                                           init];  
    locNotification.alertBody = @"You have arrived!";  
    locNotification.regionTriggersOnce = YES;  
  
    locNotification.region = [[CLCircularRegion alloc]  
                             initWithCenter:LOC_COORDINATE  
                             radius:LOC_RADIUS  
                             identifier:LOC_IDENTIFIER];
```

```
[[UIApplication sharedApplication]  
 scheduleLocalNotification:locNotification];
```


Location Notifications

Notification handling

```
- (void)application:(UIApplication *)application didReceiveLocalNotification:
    (UILocalNotification *)notification
{
    CLRegion *region = notification.region;

    if (region) {
        [self tellFriendsUserArrivedAtRegion:region];
    }
}
```

Location Notifications

Notification handling

```
- (void)application:(UIApplication *)application didReceiveLocalNotification:
    (UILocalNotification *)notification
{
    CLRegion *region = notification.region;

    if (region) {
        [self tellFriendsUserArrivedAtRegion:region];
    }
}
```

Location Notifications

Notification handling

```
- (void)application:(UIApplication *)application didReceiveLocalNotification:
    (UILocalNotification *)notification
{
    CLRegion *region = notification.region;

    if (region) {
        [self tellFriendsUserArrivedAtRegion:region];
    }
}
```

Location Notifications

Notification handling

```
- (void)application:(UIApplication *)application didReceiveLocalNotification:
    (UILocalNotification *)notification
{
    CLRegion *region = notification.region;

    if (region) {
        [self tellFriendsUserArrivedAtRegion:region];
    }
}
```

Location Notifications

Wrap up

`UINavigationController` API addition

Requires Core Location registration

`application:didReceiveLocalNotification:` not called if CoreLocation disabled

Recap

Notification actions

User and Remote Notification registration

Location Notifications

More Information

Paul Danbold

Core OS Technologies Evangelist

danbold@apple.com

Documentation

<http://developer.apple.com>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

-
- What's New in Core Location Marina Tuesday 2:00PM

 - Taking Core Location Indoors Marina Tuesday 3:15PM

 - Protecting the User's Data WWDC 2012

Labs

-
- Notifications Lab Services Lab B Wednesday 3:15PM
 - Security and Privacy Lab Core OS Lab B Thursday 3:15PM
-

 WWDC14