

Разработване на приложения за iOS



Петко Пенчев, TapBits.com

Core Location Framework

- Core Location

Framework за географско позициониране

Core Location

- Framework без UI
- Основният базов обект е CLLocation

Създава се от CLLocationManager и осигурява географски координати, височина, скорост и посока на движение на устройството

CLLocation Class

- Геоградска позиция

@property(readonly, NS_NONATOMIC_IPHONEONLY) CLLocationCoordinate2D coordinate

```
typedef struct {  
    CLLocationDegrees latitude;  
    CLLocationDegrees longitude;  
} CLLocationCoordinate2D;
```

- Точност в метри

@property(readonly, NS_NONATOMIC_IPHONEONLY) CLLocationAccuracy horizontalAccuracy
@property(readonly, NS_NONATOMIC_IPHONEONLY) CLLocationAccuracy verticalAccuracy
typedef double CLLocationAccuracy;

- Скорост в метри в секунда

@property(readonly, NS_NONATOMIC_IPHONEONLY) CLLocationSpeed speed
typedef double CLLocationSpeed;

- Посока в градуси спрямо север

@property(readonly, NS_NONATOMIC_IPHONEONLY) CLLocationDirection course
typedef double CLLocationDirection;

CLLocation Class

- Надморска височина в метри

```
@property(readonly, NS_NONATOMIC_IPHONEONLY) CLLocationDistance altitude
```

негативна стойност указва под морското равнище

- Точност в метри

```
@property(readonly, NS_NONATOMIC_IPHONEONLY) CLLocationAccuracy horizontalAccuracy
```

```
@property(readonly, NS_NONATOMIC_IPHONEONLY) CLLocationAccuracy verticalAccuracy
```

```
typedef double CLLocationAccuracy;
```

- Дата и час на извличане на резултата

```
@property(readonly, NS_NONATOMIC_IPHONEONLY) NSDate *timestamp
```

- Описание

```
- (NSString *)description
```

```
<latitude>, <longitude> +/- <accuracy>m (speed <speed> kph / heading <heading>) @ <date-time>”, where <latitude>, <longitude>, <accuracy>, <speed>, and <heading>
```

В общият случай негативна стойност означава, че стойността е невалидна - speed, course

CLLocationManager

- Как получаваме CLLocation?
чрез delegate на CLLocationManager
- CLLocationManager подходи за
ИЗПОЛЗВАНЕ
 1. Проверка за това дали хардуера, който използваме поддържа вида location service, който искаме да използваме
 2. Създаваме CLLocationManager instance и установяваме delegate клас който да получава обновленията.
 3. Конфигурираме CLLocationManager според нашите нужди и типа обновления, които искаме да получаваме.
 4. Стартираме manager-а наблюдение свързани с промени в позицията.

CLLocationManager

- Видове наблюдения свързани с промяна на географското положение, които могат да бъдат установени:
 - Обновления свързани с географска позиция, постигната точност, постоянно обновяване
 - Обновление само ако настъпи съществена промяна в позицията.
 - Обновление в случай на промяна свързана с установен регион
 - Обновления свързани с посока.

CLLocationManager

- Проверка на хардуерната поддръжка

+(BOOL)locationServicesEnabled;

Потребителя е разрешил Location Monitoring от Settings

+ (BOOL)headingAvailable;

Поддържа ли компас?

+(BOOL)significantLocationChangeMonitoringAvailable;

Може ли да известява за съществени промени в позицията. Поддържа се само ако устройството е с поддръжка на мобилна мрежа

+(BOOL)regionMonitoringAvailable;

Само някои iOS4 устройства

+(BOOL)regionMonitoringEnabled;

Разрешено от Settings

CLLocationManager

- При първо използване?
Можем да подадем въпроса към потребителя

```
@property (copy) NSString *purpose;
```

- Установяване на целева точност за наблюдение

```
@property CLLocationAccuracy desiredAccuracy;
```

```
@property CLLocationDistance distanceFilter;
```

Освен ако не целим спестяване на енергия на батерията, установяваме максимално ниски стойности.

Единствено промени в рамките на установената точност ще предизвикат събитие.

CLLocationManager

- Активиране на наблюдението за промяна

- `(void)startUpdatingLocation;`

- Деактивиране на наблюдението за промяна

- `(void)stopUpdatingLocation;`

Добрата практика е да деактивираме когато не е нужно за да се пести батерията на устройството.

CLLocationManager

- Notification чрез Delegate методи

При промяна в позицията

```
- (void)locationManager:(CLLocationManager *)manager  
  didUpdateToLocation:(CLLocation *)newLocation  
  fromLocation:(CLLocation *)oldLocation;
```

При грешка

```
- (void)locationManager:(CLLocationManager *)manager  
  didFailWithError:(NSError *)error;
```

CLLocationManager

- Наблюдение за посока

```
@property CLLocationDegrees headingFilter;
```

```
@property CLHeadingOrientation headingOrientation;
```

```
- (void)startUpdatingHeading;
```

```
- (void)stopUpdatingHeading;
```

```
- (void)locationManager:(CLLocationManager *)manager  
  didUpdateHeading:(CLHeading *)newHeading;
```

```
@property (readonly) CLLocationDirection magneticHeading; @property (readonly)  
CLLocationDirection trueHeading;
```

```
@property(readonly)CLLocationDirectionheadingAccuracy;
```

CLLocationManager

- Наблюдение за съществена промяна в позицията

‘Съществена’ не е дефинирано понятие, промяната на клетка на мобилният оператор е реалистичният критерии.

- (void)startMonitoringSignificantLocationChanges;
- (void)stopMonitoringSignificantLocationChanges;

* Работи дори когато приложението е в background

CLLocationManager

- Наблюдение за промяна в региона

- (void)startMonitoringForRegion:(CLRegion *) desiredAccuracy:(CLLocationAccuracy);
- (void)stopMonitoringForRegion:(CLRegion *);

Delegates

- (void)locationManager:(CLLocationManager *)manager
didEnterRegion:(CLRegion *)region;
- (void)locationManager:(CLLocationManager *)manager
didExitRegion:(CLRegion *)region;
- (void)locationManager:(CLLocationManager *)manager
monitoringDidFailForRegion:(CLRegion *)region
withError:(NSError *)error;

* Работи дори когато приложението е в background

Core Location Пример

- Изисква *CoreLocation.framework*
- Създаваме нов class с delegate метод

```
@protocol MyCLControllerDelegate <NSObject>
@required
- (void)locationUpdate:(CLLocation *)location;
- (void)locationError:(NSError *)error;
@end

@interface MyCLController : NSObject <CLLocationManagerDelegate> {
    CLLocationManager *locationManager;
    id delegate;
}

@property (nonatomic, retain) CLLocationManager *locationManager;
@property (nonatomic, assign) id <MyCLControllerDelegate> delegate;

- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
    fromLocation:(CLLocation *)oldLocation;

- (void)locationManager:(CLLocationManager *)manager
    didFailWithError:(NSError *)error;

@end
```

```
#import "MyCLController.h"

@implementation MyCLController

@synthesize locationManager;
@synthesize delegate;

- (id) init {
    self = [super init];
    if (self != nil) {
        self.locationManager = [[[CLLocationManager alloc] init] autorelease];
        self.locationManager.delegate = self; // send loc updates to myself
    }
    return self;
}

- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
    fromLocation:(CLLocation *)oldLocation
{
    [self.delegate locationUpdate:newLocation];
}

- (void)locationManager:(CLLocationManager *)manager
    didFailWithError:(NSError *)error
{
    [self.delegate locationError:error];
}

- (void)dealloc {
    [self.locationManager release];
    [super dealloc];
}

@end
```

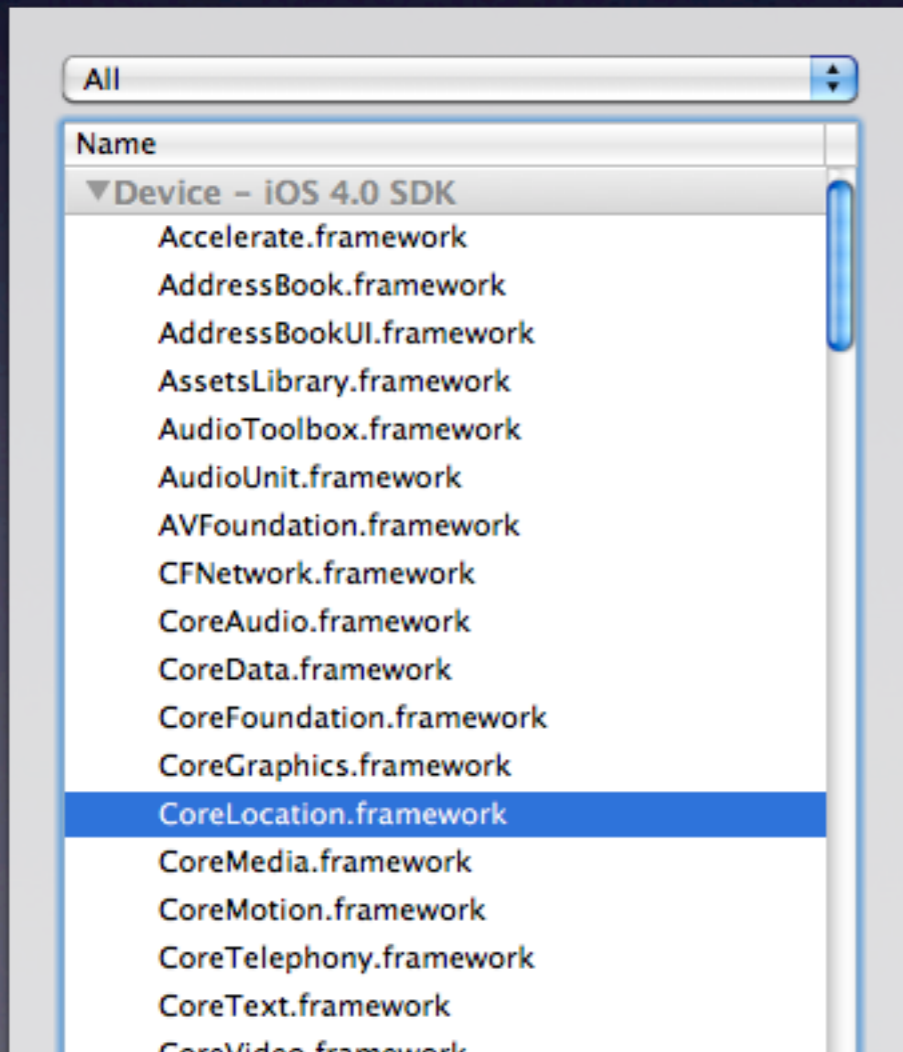
Core Location Пример

```
#import <UIKit/UIKit.h>
#import "MyCLController.h"

@interface HelloThereViewController : UIViewController <MyCLControllerDelegate> {
    IBOutlet UILabel *locationLabel;
    MyCLController *locationController;
}

- (void)locationUpdate:(CLLocation *)location;
- (void)locationError:(NSError *)error;

@end
```



```
#import "HelloThereViewController.h"

@implementation HelloThereViewController

- (void)viewDidLoad {
    locationController = [[MyCLController alloc] init];
    locationController.delegate = self;
    [locationController.locationManager startUpdatingLocation];
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)dealloc {
    [locationController release];
    [super dealloc];
}

- (void)locationUpdate:(CLLocation *)location {
    locationLabel.text = [location description];
}

- (void)locationError:(NSError *)error {
    locationLabel.text = [error description];
}

@end
```

Примери

- <http://mobileorchard.com/hello-there-a-corelocation-tutorial/>
- <http://www.vellios.com/2010/08/16/core-location-gps-tutorial/>
- http://developer.apple.com/library/ios/#documentation/CoreLocation/Reference/CoreLocation_Framework/_index.html