

Гаврилов Д.Е.

Core Location

- Права доступа
- Использование CLLocationManager
- Типы сервисов CLLocationManager и методы их делегата
- Работа в бэкграунде
- На что стоит обратить внимание

Права доступа

- «When In Use»
[self.locationManager.requestWhenInUseAuthorization]
- «Always» авторизация
[self.locationManager.requestAlwaysAuthorization]
- - (void)locationManager:(CLLocationManager *)manager
didChangeAuthorizationStatus:(CLLocationAuthorizationStatus)status

Location Services



Location Services uses GPS, Bluetooth, and crowd-sourced Wi-Fi hotspot and cell tower locations to determine your approximate location. [About Location Services & Privacy...](#)

ALLOW LOCATION ACCESS

Never

While Using the App

Always



Access to your location will be available even when this app is in the background.

Background App Refresh



Allow apps to refresh their content when on Wi-Fi or cellular in the background. Turning off apps may help preserve battery life.



News



Notes



Podcasts



Stocks



Weather



- `UIBackgroundRefreshStatus` `refreshStatus = [[UIApplication sharedApplication] backgroundRefreshStatus]`
`UIBackgroundRefreshStatusRestricted`
`UIBackgroundRefreshStatusDenied`
`UIBackgroundRefreshStatusAvailable`
- `BOOL` `isLocationServicesEnabled = [CLLocationManager locationServicesEnabled]`
- `CLAuthorizationStatus` `authorizationStatus = [CLLocationManager authorizationStatus]`
`kCLAuthorizationStatusNotDetermined`
`kCLAuthorizationStatusRestricted`
`kCLAuthorizationStatusDenied`
`kCLAuthorizationStatusAuthorizedAlways`
`kCLAuthorizationStatusAuthorizedWhenInUse`
- `BOOL` `isMonitoringAvailable = [CLLocationManager isMonitoringAvailableForClass:regionClass]`
- `BOOL` `isSignificantMonitoringAvailable = [CLLocationManager significantLocationChangeMonitoringAvailable]`
- `BOOL` `isHeadingAvailable = [CLLocationManager headingAvailable]`
- `BOOL` `isRangingAvailable = [CLLocationManager isRangingAvailable]`
- `BOOL` `isDeferredLocationUpdatesAvailable = [CLLocationManager deferredLocationUpdatesAvailable]`

Использование CLLocationManager

- Отслеживание больших или малых изменений текущего положения пользователя с выбранной точностью
- Сообщение изменений направления с компаса
- Мониторинг различных регионов, представляющих интерес, и генерация событий, когда пользователь входит или выходит из этих регионов
- Отсрочка доставки обновлений позиции когда приложение в бэкграунде
- Сообщение дальности до ближайших маячков
- Мониторинг посещений

Типы сервисов CLLocationManager

- [`self.locationManager startUpdatingLocation`]
[`self.locationManager requestLocation`]
- [`self.locationManager startMonitoringSignificantLocationChanges`]
- [`self.locationManager allowDeferredLocationUpdatesUntilTraveled:distance timeout:timeout`]

- [`self.locationManager startMonitoringForRegion:circularRegion`]
[`self.locationManager requestStateForRegion:circularRegion`]
- [`self.locationManager startRangingBeaconsInRegion:beaconRegion`]
- [`self.locationManager startMonitoringVisits`]

- `[self.locationManager startUpdatingHeading]`

CLLocationManagerDelegate

- - (void)locationManagerDidPauseLocationUpdates:(CLLocationManager *)manager
- - (void)locationManagerDidResumeLocationUpdates:(CLLocationManager *)manager

- - (void)locationManager:(CLLocationManager *)manager
didUpdateLocations:(NSArray<CLLocation *> *)locations
- - (void)locationManager:(CLLocationManager *)manager
didFailWithError:(NSError *)error
- - (void)locationManager:(CLLocationManager *)manager
didFinishDeferredUpdatesWithError:(nullable NSError *)error

- - (void)locationManager:(CLLocationManager *)manager
didStartMonitoringForRegion:(CLRegion *)region
- - (void)locationManager:(CLLocationManager *)manager
didEnterRegion:(CLRegion *)region
- - (void)locationManager:(CLLocationManager *)manager
didExitRegion:(CLRegion *)region
- - (void)locationManager:(CLLocationManager *)manager
didDetermineState:(CLRegionState)state
forRegion:(CLRegion *)region
- - (void)locationManager:(CLLocationManager *)manager
monitoringDidFailForRegion:(nullable CLRegion *)region
withError:(NSError *)error

- - (void)locationManager:(CLLocationManager *)manager
didRangeBeacons:(NSArray<CLBeacon *> *)beacons
inRegion:(CLBeaconRegion *)region
- - (void)locationManager:(CLLocationManager *)manager
rangingBeaconsDidFailForRegion:(CLBeaconRegion *)region
withError:(NSError *)error

- - (void)locationManager:(CLLocationManager *)manager
didUpdateHeading:(CLHeading *)newHeading
- - (void)locationManager:(CLLocationManager *)manager
didVisit:(CLVisit *)visit

Работа в бэкграунде

- Использовать сервисов геолокации в бэкграунде должно быть обосновано.
- Можно использовать стандартный сервис геолокации в бэкграунде, если приложение предоставляет сервис, которому необходим постоянное обновление геолокации. Например фитнес или навигационное приложение.
- Когда приложение перезапускается сервисом геолокации то в `launchOptions` в методах `application:willFinishLaunchingWithOptions:` и `application:didFinishLaunchingWithOptions:` будет содержаться ключ `UIApplicationLaunchOptionsLocationKey`.
- Если приложение было завершено пользователем или системой, то для продолжения обновления геолокации нужно, чтобы пользователь запустил приложение самостоятельно при использовании стандартного сервиса геолокации.
- Нужно помнить, что пользователь может отключить бэкграунд обновления совсем или для нашего приложения в настройках.

API	Requires location updates in background mode	Can use location updates in background mode
<code>startUpdatingLocation</code>	Yes	Yes
<code>CLVisit</code>	No	Yes
<code>CLRegion</code>	No	Yes
<code>startMonitoringSignificantLocationChanges</code>	No	Yes
<code>CLBeaconRegion</code>	No	Yes, on a case-by-case basis

На что стоит обратить внимание

- **CLLocationManager**

@property(assign, nonatomic) CLLocationDistance distanceFilter;

@property(assign, nonatomic) CLLocationAccuracy desiredAccuracy;

@property(assign, nonatomic) BOOL pausesLocationUpdatesAutomatically

@property(assign, nonatomic) BOOL allowsBackgroundLocationUpdates

@property(assign, nonatomic) CLActivityType activityType;

CLActivityTypeOther = 1,

CLActivityTypeAutomotiveNavigation,

CLActivityTypeFitness,

CLActivityTypeOtherNavigation

- **CLLocation**

@property(readonly, nonatomic, copy) NSDate *timestamp;

- **Info.plist**

NSLocationWhenInUseUsageDescription

NSLocationAlwaysUsageDescription