

# iOS的三种常见计时器（NSTimer、CADisplayLink、dispatch\_source\_t）的使用



爱笑的猫mi

关注

2019.08.15 13:40:53 字数 1,132 阅读 2,215

## 一、介绍

在iOS中，计时器是比较常用的，用于统计累加数据或者倒计时等，例如手机号获取验证码。计时器大概有那么三种，分别是：NSTimer、CADisplayLink、dispatch\_source\_t

## 二、使用

```
1 @property (strong, nonatomic) NSTimer *timer;
2 @property (strong, nonatomic) CADisplayLink *displayLinkTimer;
3 @property (strong, nonatomic) dispatch_source_t sourceTimer;
```

### 1、NSTimer:

```
1 //NSTimer
2 -(void)createTimer{
3
4     //初始化
5     //__timer = [NSTimer scheduledTimerWithTimeInterval:1.0 repeats:YES block:^(NSTimer
6     //执行操作
7     //};
8
9     __timer = [NSTimer scheduledTimerWithTimeInterval:1.0 target:self selector:@selector
10
11     //加入RunLoop循环池
12     [[NSRunLoop mainRunLoop] addTimer:__timer forMode:NSDefaultRunLoopMode];
13
14     //开启定时器
15     [__timer fire];
16 }
```

```
1 -(void)timerStart:(NSTimer *)timer{
2     NSLog(@"%s-----%lf",__func__,__timer.timeInterval);
3
4     //每隔定时间
5     [__timer invalidate];
6     //__timer = nil;
7 }
```

解释：

TimeInterval: 执行之前等待的时间。比如设置成1.0，就代表1秒后执行方法

target: 需要执行方法的对象。

selector: 需要执行的方法

repeats: 是否需要循环

注意：

调用创建方法后，target对象的计数器会加1，直到执行完毕，自动减1。如果是循环执行的话，就必须手动关闭，否则可以不执行释放方法。

特性：

存在延迟

不管是一次性的还是周期性的timer的实际触发事件的时间，都会与所加入的RunLoop和RunLoop Mode有关，如果此RunLoop正在执行一个连续性的运算，timer就会被延时出发。重复性的timer遇到这种情况，如果延迟超过了一个周期，则会在延时结束后立刻执行，并按照之前指定的周期继续执行。

必须加入RunLoop

使用上面的创建方式，会自动把timer加入MainRunLoop的NSDefaultRunLoopMode中。如果使用以下方式创建定时器，就必须手动加入RunLoop:

```
1 NSTimer *timer = [NSTimer timerWithTimeInterval:5 target:self selector:@selector(timer
2 [[NSRunLoop mainRunLoop] addTimer:timer forMode:NSDefaultRunLoopMode];
```

### 2、CADisplayLink:

```
1 //CADisplayLink
2 -(void)createCADisplayLink{
3
4     __displayLinkTimer = [CADisplayLink displayLinkWithTarget:self selector:@selector(h
5     [__displayLinkTimer addToRunLoop:[NSRunLoop currentRunLoop] forMode:NSDefaultRunLo
6 }
```

```
1 -(void)handleDisplayLink:(CADisplayLink *)displayLinkTimer{
2     NSLog(@"%s-----%ld",__func__,__displayLinkTimer.preferredFramesPerSecond);
3
4     //每隔定时间
5     [__displayLinkTimer invalidate];
6     //__displayLinkTimer = nil;
7 }
```

解释：

当把CADisplayLink对象add到runloop中后，selector就能被周期性调用，类似于重复的NSTimer被启动了；执行invalidate操作时，CADisplayLink对象就会从runloop中移除，selector调用也随即停止，类似于NSTimer的invalidate方法。

特性：

```
1 屏幕刷新时调用CADisplayLink是一个能让我们以和屏幕刷新率同步的频率将特定的内容画到屏幕上的定时类别。C
2 延迟iOS设备的屏幕刷新率是固定的，CADisplayLink在正常情况下会在每次刷新结束都被调用，精度相当高。
```

使用场景：

从原理上可以看出，CADisplayLink适合做界面的不停重绘，比如视频播放的时候需要不停地获取下一帧用于界面渲染。

重要属性：

frameInterval(已过时，用preferredFramesPerSecond替代) NSInteger类型的值，用来设置间隔多少帧调用一次selector方法，默认值是1，即每帧都调用一次。

duration readOnly的CFTimeInterval值，表示两次屏幕刷新之间的时间间隔。需要注意的是，该属性在target的selector被首次调用以后才会被赋值。selector的调用间隔时间计算方式是：调用间隔时间 = duration × frameInterval。

### 3、dispatch\_source\_t:

```
1 //dispatch_source_t
2 -(void)createDispatch_source_t{
3
4     //创建全局队列
5     dispatch_queue_t queue = dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT
6
7     //使用全局队列创建计时器
8     __sourceTimer = dispatch_source_create(DISPATCH_SOURCE_TYPE_TIMER, 0, 0, queue);
9
10    //定时延迟时间
11    NSTimeInterval delayTime = 1.0f;
12
13    //定时时间间隔时间
14    NSTimeInterval timeInterval = 1.0f;
15
16    //设置开始时间
17    dispatch_time_t startDelayTime = dispatch_time(DISPATCH_TIME_NOW, (int64_t)(delayT
18
19    //设置计时器
20    dispatch_source_set_timer(__sourceTimer, startDelayTime, timeInterval * NSEC_PER_SEC, 0
21
22    //执行事件
23    dispatch_source_set_event_handler(__sourceTimer, ^{
24
25        //销毁定时器
26        dispatch_source_cancel(__myTimer);
27    });
28
29    //启动计时器
30    dispatch_resume(__sourceTimer);
31 }
```

特性：

默认是重复执行的，可以在事件响应回调中通过dispatch\_source\_cancel方法来设置为只执行一次，如下代码：

```
1 dispatch_source_set_event_handler(__timer, ^{
2     //执行事件
3     dispatch_source_cancel(__timer);
4 });
```

重要属性：

```
1 dispatch_source_set_timer(dispatch_source_t source,
2 dispatch_time_t start,
3 uint64_t interval,
4 uint64_t leeway);
```

- start计时器起始时间，可以通过dispatch\_time创建，如果使用DISPATCH\_TIME\_NOW，则创建后立即执行
- interval计时器间隔时间，可以通过timeInterval \* NSEC\_PER\_SEC来设置，其中，timeInterval为对应的秒数
- leeway这个参数的理解，我觉得<http://www.dreamingwish.com>上Seven's同学的解释很直观也很易懂：“这个参数告诉系统我们需要计时器触发的精准程度。所有的计时器都不会保证100%精准，这个参数用来告诉系统你希望系统保证精准的努力程度。如果你希望一个计时器没五秒触发一次，并且越准确越好，那么你传递0为参数。另外，如果是一个周期性任务，比如检查email，那么你会希望每十分钟检查一次，但是不用那么精准。所以你可以传入60，告诉系统60秒的误差是可接受的。这样有什么意义呢？简单来说，就是降低资源消耗。如果系统可以让cpu休息足够长的时间，并在每次醒来的时候执行一个任务集合，而不是不断的醒来睡去以执行任务，那么系统会更高效。如果传入一个比较大的leeway给你的计时器，意味着你允许系统拖延你的计时器来将计时器任务与其他任务联合起来一起执行。

优点：

时间准确

可以使用子线程，解决决定时间跑在主线程上卡UI问题

注意事项：

需要将dispatch\_source\_t timer设置为成员变量，不然会立即释放

0人点赞 > 日记本 ...

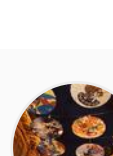
更多精彩内容，就在简书APP



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



爱笑的猫mi

总资产8 共写了8.7W字 获得94个赞 共28个粉丝

关注

## 什么是电商仓储



## 被以下专题收入，发现更多相似内容

+ 收入我的专题

## 推荐阅读

更多精彩内容 >

### 2018-05-iOS：三种常见计时器（NSTimer、CADisplayLink、dispa...

一、介绍 在iOS中，计时器是比较常用的，用于统计累加数据或者倒计时等，例如手机号获取验证码。计时器大概有那么三种...

\_coCa\_ 阅读 288 评论 0 赞 0

### iOS定时器NSTimer、CADisplayLink、dispatch\_source\_t以及...

1. 简介 iOS常用的计时器大概有三种，分别是：NSTimer、CADisplayLink、dispatch...

Timir 阅读 753 评论 0 赞 6

### 猎头公司是如何招聘和收费的



### iOS三种定时器的用法NSTimer、CADisplayLink、GCD

一、NSTimer1.创建方法 TimeInterval: 执行之前等待的时间。比如设置成1.0，就代表1秒后...

心至静行至通 阅读 1,297 评论 0 赞 5

### 3.作用域和闭包

//-----题目----->1.说一下对变量提...

飞菲fly 阅读 267 评论 0 赞 0



### 助力孩子成长日记第83天

2019年1月28日 星期一 天气晴 今天是星期一也是小年。早上，我早早起来去做早饭。我给孩子下的馄饨，我下好馄饨去...

助力孩子成长日记周晓然妈妈 阅读 38 评论 0 赞 0

## 推荐阅读

iOS 定时器

NSTimer+dispatchTimer.md

阅读 88

iOS总结篇-RunLoop

阅读 100

Android常问的面试题(一)

阅读 483

JavaScript 定时器

阅读 148

Unity 简单计时器Timer

阅读 126



动漫设计公司