

iOS (<http://lib.csdn.net/base/ios>)

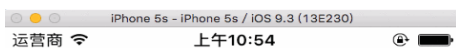
iOS (<http://lib.csdn.net/base/ios>) - 企业级开发 (<http://lib.csdn.net/ios/node/686>) - 按需加载 (<http://lib.csdn.net/ios/knowledge/1495>)

👁 106 💬 1

## ios新手开发——toast提示和旋转图片加载框

作者：qq\_16674697 ([http://my.csdn.net/qq\\_16674697](http://my.csdn.net/qq_16674697))

不知不觉自学ios已经一个月了,从OC语法到app开发,过程虽然枯燥无味,但是结果还是挺有成就感的,在此分享我的ios开发之路中的小小心得~废话不多说,先上我们今天要实现的效果图:



旋转图片loading

系统菊花loading

---

有过一点做APP经验的都知道,提示框和等待加载框一直是APP首当其中的效果,ios不像android一样,自带toast和progressbarDialog,所以在做ios开发的时候,我首先想到了先封装这两个基础控件~当然网上的资源数不胜数,但是博主抱着一颗自主研究的精神,做出的效果也不错,也已适配了所有iphone型号和版本.望大家多多支持~

## YPXToastView实现

接触过安卓开发的ios开发者可能对待toast这么个东西很不陌生,它主要是一种轻量级的提示,代替了复杂的对话框,有的显示在中间,有的显示在屏幕下方,当然,这些都是根据需求而来的.废话不多说,首先清理一下我们实现这个toast的一些必要思路:

- 1.实现的基础控件-----UILabel封装
- 2.弹出的时间和透明度变化设置
- 3.显示的位置调整

### 一.UILabel的封装

首先我们想要实现一下这个效果,首当其冲的肯定想到UILabel,那么接下来就是对UILabel的封装了,首先我们创建一文件继承UILabel,然后写好要对外暴露的方法:

```
@interface YPXToastView : UILabel

{
    @public
    CGFloat screenWidth,screenHeight;
    int _corner;
    int _duration;
}

@property(assign,nonatomic)int corner;
@property(assign,nonatomic)int duration;

-(void)showToastViewWithText:(NSString *)text andDuration:(int)duration andParentView:(UIView *)parentView;

-(void)showToastViewWithText:(NSString *)text andParentView:(UIView *)parentView;

-(void)showToastViewWithText:(NSString *)text andDuration:(int)duration andCorner:(int)corner andParentView:(UIView *)parentView;

+(void)showToastViewWithText:(NSString *)text andDuration:(int)duration andParentView:(UIView *)parentView;

+(void)showToastViewWithText:(NSString *)text andParentView:(UIView *)parentView;

+(void)showToastViewWithText:(NSString *)text andDuration:(int)duration andCorner:(int)corner andParentView:(UIView *)parentView;

-(void)setBackgroundWithColor:(UIColor *)color;

@end
```

定义了四个全局变量,两个属性,分别制定了提示框的圆角和时间.方法中定义了两个类方法,和四个实例方法,主要是因为我们在使用时并不想实例化一次我们的提示框,所有的实例方法中抽出了三个类方法方便用户调用.

下面我们来看内部主要方法实现:

```
/**
 * 新建UI
 *
 * @param str 要显示的文本
 */
-(void)createUIByText:(NSString *)str{
    self.textAlignment = NSTextAlignmentCenter;
    self.backgroundColor = [UIColor colorWithRed:00 green:00 blue:00 alpha:0.5];
    self.alpha = 0.8;
    self.text=str;
    self.font = [UIFont systemFontOfSize:14];
    self.textColor=[UIColor whiteColor];
    NSDictionary *attributes = @{NSFontAttributeName:[UIFont systemFontOfSize:self.font.pointSize],};
    CGSize textSize = [self.text boundingRectWithSize:CGSizeMake(100, 100) options:NSStringDrawingTruncatesLastVisibleLine attributes:attributes context:nil].size;;
    self.frame=CGRectMake(screenWidth/2-(textSize.width*1.7)/2, screenHeight*0.5,textSize.width*1.7,
                           textSize.height*2);
    self.layer.cornerRadius = _corner;
    self.clipsToBounds = YES;
}

-(void)setBackgroundWithColor:(UIColor *)color{
    self.backgroundColor =color;
}

/**
 * 初始化测量数据
 */
-(void)caculateSize{
    screenWidth=[UIScreen mainScreen].bounds.size.width;
    screenHeight=[UIScreen mainScreen].bounds.size.height;
}
```

方法一目了然,指定了UILabel的居中方式和背景,并设置属性让其宽度自适应,涉及到一些简单的frame计算,主要是定位于屏幕中间,宽度设为文本宽度的1.7倍,看起来比较适中.y点主要就是屏幕高度的一半,理应减去文本的高度的一半,但是博主在这偷个懒,并没有计算label的高度,所以就不赘述了~~

## 二.弹出的时间和透明度变化设置

原理很简单,就是设定了一个animateWithDuration的block回调,然后设置label的透明度和时间,具体实现如下:

```
/**
 * 显示toast
 *
 * @param parentView <#parentView description#>
 */
-(void)showToastByParentView:(UIView *)parentView{
    [parentView addSubview:self];
    //animateWithDuration可以控制label显示持续时间
    [UIView animateWithDuration:_duration animations:^(
        self.alpha = 1.0;
    ) completion:^(BOOL finished){
        [self removeFromSuperview];
    }];
}
```

默认时间为1秒,思路很清晰,先添加进我们的parentView中,然后指定时间后移除.

到此,我们的YPXToastView已经全部完成,其实内部逻辑主要是对UILabel的定制,思路简单,但是对于ios开发之路的封装思想有很大的帮助.调用时只需要一行代码:

```
[YPXToastView showToastViewWithText:@"已开启" andDuration:3 andCorner:5 andParentView:self.view];
```

调用方便简洁,以后测试就不需要用NSLog了嘿嘿~

## YPXLoddingView实现

相信在ios的开发中少不了加载等待框的开发,毕竟原生系统中貌似没有这样的对话框,我们在访问网络或者读取数据时可能需要给用户一个等待回馈,这里就用到了我们的等待加载.上面的gif中提供了两种等待加载框的样式,一种是自定义图片的旋转,顺时针或者逆时针,另一种是使用系统的UIActivityIndicatorView,使用大的加载Loading.具体开发思路如下:

- 1.继承UIView通过添加UIImageView和UILabel来组合实现
- 2.控制UIImageView的旋转以及UILabel的三个点的动态效果
- 3.显示和隐藏

### 一.UIView的封装

通过效果我们可以一目了然的知道,实现这个控件至少需要一个UIImageView(或者UIActivityIndicatorView)和UILabel,一个提供加载图片,一个提供加载文本,组合方式为竖直方向,然后设置背景的透明度.具体.h文件如下:

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

@interface YPXLoadingView : UIView

{
    @public
    int num;
    CGFloat angle;
    BOOL isShowLoading;
    UIImageView * imageView;
    UILabel * label;
    CGFloat width;
    CGFloat x;
    CGFloat y,screenWidth,screenHeight;
    UIView * _parentView;
    NSString * _text;
    NSTimer * _timer;
    UIActivityIndicatorView * _activityView;
    UIView * view;
}

@property(retain, nonatomic) NSTimer * timer;
@property(copy, nonatomic) NSString * text;
@property(retain, nonatomic) UIActivityIndicatorView * activityView;

-(void)showLoadingViewWithText:(NSString *) string;

-(void)dismissLoadingView;

-(instancetype)initWithParentView:(UIView *) parentView;

+(id)initWithParentView:(UIView *) parentView;

-(BOOL)isShowing;

-(void)showLoadingView;

-(void)showLoadingViewWithStyle:(int)style;

-(void)showLoadingViewWithText:(NSString * )text andStyle:(int)style;

@end
```

定义了一些必要的属性,包括计时器和显示文本等,主要功能为show开头的方法,style应该是个枚举类型,但是博主目前还没有写过枚举类,所以直接引用0和1来指定使用图片还是系统的菊花加载.看完.h我们来看看具体的UIView代码实现:

```
/**
 * 计算一些必要尺寸
 */
@param parentView <#parentView description#>
*/
-(void)caculatSizeWithTarget:(UIView *) parentView
{
    screenWidth=[UIScreen mainScreen].bounds.size.width;
    screenHeight=[UIScreen mainScreen].bounds.size.height;
    width=screenWidth*0.3;
    x= screenWidth/2-width/2;
    y= screenHeight/2-width/2;
    angle=0;
    num=0;
    isShowLoading=N0;
    _parentView=parentView;
}

/**
 * 创建loading视图
 */
-(void)creatLoadingView
{
    view=[[UIView alloc]init];
    view.frame=CGRectMake(0, 0, screenWidth, screenHeight);

    imageView=[[UIImageView alloc]init];
    imageView.frame=CGRectMake(width/2-width*0.5/2,15, width*0.5,width*0.4);
    imageView.clipsToBounds=YES;
    imageView.layer.rasterizationScale=[UIScreen mainScreen].scale;
    [imageView setImage:[UIImage imageNamed:@"loading.png"]];

    _activityView=[[UIActivityIndicatorView alloc]initWithFrame:CGRectMake(width/2-width*
0.55/2,15, width*0.55,width*0.45)];
    _activityView.activityIndicatorViewStyle=UIActivityIndicatorViewStyleWhiteLarge;

    label=[[UILabel alloc]init];
    label.textColor=[UIColor whiteColor];
    label.font=[UIFont systemFontOfSize:14];
    int y2=imageView.frame.size.height+(width-imageView.frame.size.height)/2;
    label.frame=CGRectMake(0,y2, width, 20);
    label.textAlignment=NSTextAlignmentCenter;
}
```

手动布局,我们指定了imageView和label的frame,通过一系列计算,把imageView设为UIView中上部,并留出四周的边距,看起来更亲切自然一点.label的位置根据imageView的frame来指定,这样就可以完成适配避免在不同屏幕上显示不同的问题.完场上述代码,一个

初步的静态效果已经生成,剩下的就是添加动画;

## 二.UIImageView旋转动画以及UILabel点点动态展示

imageview的动画添加很简单,因为我们只是涉及一点点的旋转动画,其中并没有加速度变化,读者若是想要添加,可以自己尝试一下.旋转动画的实现方式有两种:

一种是用animateWithDuration来动态的旋转一定角度,然后通过延时来改变旋转的速率,好处是简单,但是缺点也很明显,在5s中动画显得僵硬,并伴随着一点点的卡顿,如下是第一种动画方案的代码:

```
/**
 * 开启loading动画
 */
- (void)startAnimation
{
    if(isShowLoading==YES){
        CGAffineTransform endAngle = CGAffineTransformMakeRotation(angle * (M_PI / -180.0
f));
        [UIView animateWithDuration:0.03f delay:0 options:UIViewAnimationOptionCurveLinear animations:^(
            imageView.transform =endAngle;
        } completion:^(BOOL finished) {
            if(angle==360){
                angle=0;
            }
            if(angle==0||angle==360){
                label.text=[_text stringByAppendingString:@"..."];
            }else if(angle==90){
                label.text=_text;
            }else if(angle==180){
                label.text=[_text stringByAppendingString:@"."];
            }else if(angle==270){
                label.text=[_text stringByAppendingString:@".."];
            }
            angle += 10;

            [self startAnimation];
        }]);
    }
}
```

通过改变imageview的角度来旋转图片的方式,使用block回调中的角度关系,我们可以动态的设置提示文本省略号的动态展示.因为实现效果有点卡顿,所以博主采用了第二种实现



方式,代码如下:

```
/**
 * 启动计数定时器
 */
-(void)UpdateText
{
    num++;
    if (num>4) {
        num=0;
    }
    if(num==0||num==4){
        label.text=[_text stringByAppendingString:@"..."];
    }else if(num==1){
        label.text=_text;
    }else if(num==2){
        label.text=[_text stringByAppendingString:@"."];
    }else if(num==3){
        label.text=[_text stringByAppendingString:@".."];
    }
}

/**
 * 给imageView添加动画
 *
 * @param imageView imageview
 *
 * @return imageview
 */
+ (UIImageView *)rotateImageView:(UIImageView *)imageView
{
    CABasicAnimation *animation = [ CABasicAnimation
                                    animationWithKeyPath: @"transform" ];
    animation.fromValue = [NSValue valueWithCATransform3D:CATransform3DIdentity];

    //围绕Z轴旋转, 垂直与屏幕
    animation.toValue = [ NSValue valueWithCATransform3D:
                           CATransform3DMakeRotation(M_PI, 0.0, 0.0, 1.0) ];
    animation.duration = 0.5;
    //旋转效果累计, 先转180度, 接着再旋转180度, 从而实现360旋转
    animation.cumulative = YES;
    animation.repeatCount = 10000;

    [imageView.layer addAnimation:animation forKey:nil];
    return imageView;
}
```

采用CABasicAnimation的动画效果可以达到动画流畅度的完美展示,优点就是增加了旋转性能,缺点就是没有像animateWithDuration那样有动画的回调,这样我们就没有办法动态的去改变label的提示文本,所以细心的读者会发现,博主前面的.h文件中已经声明了一个定时器,那么这个定时器的作用是用来干嘛的呢?我们通过启动定时器,来动态的刷新label

的提示文本达到一种动态展示的效果,这种思路在安卓里也同样适用.

完成了我们的图片旋转,基本上这个功能已经完成了百分之八十,剩下就是显示和隐藏了;

### 三.显示和隐藏

前面介绍.h文件声明的时候,已经把本控件的所有调用方法已经列出来了,其中包含了一系列的.show方法,因为loading这种控件,我们可能需要对其状态进行判断,而且可能在网络请求中调用多次,为了不浪费内存,我们在这里提倡使用单例模式,并初始化一个Loading在ViewDidLoad中.后期调用只需要show和dismiss即可,下面我们来看具体的show和dismiss的方法实现:

```
/**
 * 显示loading.默认文本为 "正在加载"
 */
-(void)showLoadingView
{
    if(isShowLoading==YES){
        return;
    }
    if(_text==nil||[_text isEqualToString:@""]){
        _text=@"正在加载";
    }
    label.text=_text;
    isShowLoading=YES;
    angle=0;
    self.hidden=NO;
    [self addSubview:imageView];
    [self addSubview:label];
    [view addSubview:self];
    [_parentView addSubview:view];
    [YPXLoadingView rotateImageView:imageView];
    _timer=[NSTimer scheduledTimerWithTimeInterval:0.5 target:self selector:@selector(UpdateText) userInfo:nil repeats:YES];
}

-(void)showLoadingViewWithStyle:(int)style
{
    if(style==0){//菊花加载
        if(isShowLoading==YES){
            return;
        }
        if(_text==nil||[_text isEqualToString:@""]){
            _text=@"正在加载";
        }
        label.text=_text;
        isShowLoading=YES;
        angle=0;
        self.hidden=NO;
        [self addSubview:_activityView];
    }
}
```

```

        [self addSubview:label];
        [imageView removeFromSuperview];
        [_activityView startAnimating];
        [view addSubview:self];
        [_parentView addSubview:view];
        _timer=[NSTimer scheduledTimerWithTimeInterval:0.5 target:self selector:@selector
(UpdateText) userInfo:nil repeats:YES];

    }else{//旋转图片加载
        [self showLoadingView];
    }
}

/**
 * 显示loading
 *
 * @param string 显示的文本
 */
-(void)showLoadingViewWithText:(NSString *) string
{
    _text=string;
    [self showLoadingView];
}

-(void)showLoadingViewWithText:(NSString *)text andStyle:(int)style{
    _text=text;
    [self showLoadingViewWithStyle:style];
}

```

```

/**
 * 消失loading
 */
-(void)dismissLoadingView
{
    self.hidden=YES;
    isShowLoading=NO;
    [_timer invalidate];
    [imageView.layer removeAllAnimations];
    [_activityView stopAnimating];
    [view removeFromSuperview];
}

```

总体来说show方法中就是单纯的控制了imageView和\_activityView通过style来隐藏和显示,思路很简单,再次不做赘述.dismiss中只需要移除我们的view就好,非常简单,同时不要忘记stop我们的\_activityView以及关闭定时器就好.

至此,所有的代码实现已经完成,我们在需要调用的地方首先实例化一次,然后使用show和dismiss即可.

## 总结

ios开发总体来说还算顺风顺水,因为对安卓有一定的基础,学习oc等面向对象的语法不免要快一点,但是ios中对于控件的方法并不是很多,甚至某些安卓一行代码就能实现的功能,ios需要好多行,这就是一个语言的魅力所在,当然,在自学ios的过程中我会不断的通过写博客的方式来提升自己的水平,在新手开发道路中,希望我能雨你们同行,谢谢读者的支持  
~~ 😊

下载地址 : [http://download.csdn.net/detail/qq\\_16674697/9622230](http://download.csdn.net/detail/qq_16674697/9622230)

([http://download.csdn.net/detail/qq\\_16674697/9622230](http://download.csdn.net/detail/qq_16674697/9622230))

作者 : yangpeixing

QQ : 313930500

转载请注明出处~谢谢~

[查看原文>> \(http://blog.csdn.net/qq\\_16674697/article/details/52437724\)](http://blog.csdn.net/qq_16674697/article/details/52437724)



0

---

### 看过本文的人也看了 :

- iOS知识结构图  
(<http://lib.csdn.net/base/ios/structure>)

- iOS中 UIWebView加载网络数据 技术分...  
(<http://lib.csdn.net/article/ios/36943>)

- ionic入门教程第十九课-ionic路由详解 ( ...  
(<http://lib.csdn.net/article/ios/38649>)
- iOS网络编程(三) 异步加载及缓存图片...  
(<http://lib.csdn.net/article/ios/36941>)
- iOS UITableViewCell 多线程 网络+沙...  
(<http://lib.csdn.net/article/ios/36940>)
- iOS APP中数据加载的6种方式  
(<http://lib.csdn.net/article/ios/36937>)

## 发表评论

输入评论内容

发表

## 1个评论



(<http://my.csdn.net/u011733020>)

**u011733020** (<http://my.csdn.net/u011733020>)

不错，很详细，继续努力。

2016-09-19 09:12:45

回复

公司简介 (<http://www.csdn.net/company/about.html>) | 招贤纳士 (<http://www.csdn.net/company/recruit.html>) |  
广告服务 (<http://www.csdn.net/company/marketing.html>) | 银行汇款帐号 (<http://www.csdn.net/company/account.html>)  
| 联系方式 (<http://www.csdn.net/company/contact.html>) | 版权声明 (<http://www.csdn.net/company/statement.html>) |  
法律顾问 (<http://www.csdn.net/company/layer.html>) | 问题报告 (<mailto:webmaster@csdn.net>) |  
合作伙伴 (<http://www.csdn.net/friendlink.html>) | 论坛反馈 (<http://bbs.csdn.net/forums/Service>)

网站客服 杂志客服 (<http://wpa.qq.com/msgrd?v=3&uin=2251809102&site=qq&menu=yes>)

微博客服 (<http://e.weibo.com/csdnsupport/profile>) webmaster@csdn.net (<mailto:webmaster@csdn.net>) 400-600-2320 |

北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved

 (<http://www.hd315.gov.cn/beian/view.asp?bianhao=010202001032100010>)