

UITextView, UITextField 中有很多坑,网上的方法也很多,但是用过之后暂时没有发现一个好用。这里我给大家几组测试用例可以一试,为啥不好用。

1. 限制10个字节,输入2个Emoj之后是8个字节(一个Emoj是4个字节),此时再输入一个中文,看看结果如何(中文的UTF8占3个字节)
2. 限制5个字符,一个Emoj是2个字符,其他都是一个。此时输入两个Emoj,再输入中文,然后中文联想试试。

就目前的情况来说,看了很多资料,并没有一个通用的能限制字符数和字节数的封装。这里全面进行了总结,并进行了封装。欢迎大家下载。

一. 字符限制

1. 错误方法

常见的这种方法是错误的,会导致Emoj表情的截取问题

```
- (BOOL)textField:(UITextField *)textField
shouldChangeCharactersInRange:(NSRange)range replacementString:
(NSString *)string
{
    if(range.length + range.location > textField.text.length)
    {
        return NO;
    }

    NSUInteger newLength = [textField.text length] + [string
length] - range.length;
    return newLength <= 5;
}
```

这种限制方法会导致拼音下出现这种情况,且无法输入.无法输入满5个字符。在emoj表情也有问题

姓名

xiang

姓名

向晨宇ze

1 泽

2 则

3 择

4 责



姓名



2. 推荐方法 使用`rangeOfComposedCharacterSequencesForRange`, 防止在`range`范围内整词被截断.因为中文的UTF8是3字节, Emoj是4个字节, 且不能边输入边限制, 否则中文联想的时候就无法继续输入。只能输入后在`textFieldDidChange`的时候进行截断。综上所述, 思路如下:

```
- (void)textFieldDidChange:(UITextField *)textField
{
    NSString *text = textField.text;
    // NSLog(@"text:%@", text);

    UITextRange *selectedRange = [textField markedTextRange];
    UITextPosition *position = [textField
positionFromPosition:selectedRange.start offset:0];

    // 没有高亮选择的字, 则对已输入的文字进行字数统计和限制, 防止中文被截断

    if (!position){
        //---字符处理
        if (text.length > _maxLength){
            // 中文和emoji表情存在问题, 需要对此进行处理
            NSRange range;
            NSUInteger inputLength = 0;
            for(int i=0; i < text.length && inputLength <=
_maxLength; i += range.length) {
                range = [textField.text
rangeOfComposedCharacterSequenceAtIndex:i];
                inputLength += [text
substringWithRange:range].length;
                if (inputLength > _maxLength) {
                    NSString* newText = [text
substringWithRange:NSMakeRange(0, range.location)];
                    textField.text = newText;
                }
            }
        }
    }
}
```

```
}
```

姓名

密码

ResignFirst

RejectKeyboard

nextController

大大方方发个广告 大大方方 鼎鼎 ^

二. 字节限制

1. 限制字节数

在UTF8中，英文和数字是1个字节,汉子是3个字节，emoji是3或者4个字节。这里的难度比上面更大，如果截取失败，极有可能出现乱码。这里我们的做法如下

```
- (void)textFieldDidChange:(UITextField *)textField
{
    NSString *text = textField.text;
    // NSLog(@"text:%@", text);

    UITextRange *selectedRange = [textField markedTextRange];
    UITextPosition *position = [textField
positionFromPosition:selectedRange.start offset:0];

    // 没有高亮选择的字，则对已输入的文字进行字数统计和限制，防止中文被截断
```

```

    if (!position){
        //---字节处理
        //Limit
        NSUInteger textBytesLength = [textField.text
lengthOfBytesUsingEncoding:NSUTF8StringEncoding];
        if (textBytesLength > _maxBytesLength) {
            NSRange range;
            NSUInteger byteLength = 0;
            for(int i=0; i < text.length && byteLength <=
_maxBytesLength; i += range.length) {
                range = [textField.text
rangeOfComposedCharacterSequenceAtIndex:i];
                byteLength += strlen([[text
substringWithRange:range] UTF8String]);
                if (byteLength > _maxBytesLength) {
                    NSString* newText = [text
substringWithRange:NSMakeRange(0, range.location)];
                    textField.text = newText;
                }
            }
        }
        if (self.textFieldChange) {
            self.textFieldChange(self, textField.text);
        }
    }
}

```

三. 放弃键盘

1. 能拿到UITextField的时候用

```

- (BOOL)textFieldShouldReturn:(UITextField *)textField
{
    return [textField resignFirstResponder];
}

```

2. 点击view消失的时候用

```

[self.view endEditing:YES];

```

3. 难以获取的时候用

```
[[UIApplication sharedApplication]
sendAction:@selector(resignFirstResponder) to:nil from:nil
forEvent:nil];
```

或者

```
[[[UIApplication sharedApplication] keyWindow] endEditing:YES];
```

4. Tableview点击空白处或者滚动时消失

```
{
    UITapGestureRecognizer *singleTap = [[UITapGestureRecognizer
alloc] initWithTarget:self action:@selector(fingerTapped:)];
    [self.view addGestureRecognizer:singleTap];
}

#pragma mark- 键盘消失
-(void)fingerTapped:(UITapGestureRecognizer *)gestureRecognizer{
    [self.view endEditing:YES];
}
-(void)scrollViewWillBeginDragging:(UIScrollView *)scrollView{
    [self.view endEditing:YES];
}
```

四. 正则表达式限制

请参考[正则表达式语法表](#),这里我提供了两种表达式给大家参考, 一个Int,一个无unsignedInt

```
-(BOOL) isTextFieldMatchWithRegularExpression:(NSString
*)expression{

    NSPredicate *predicate = [NSPredicate
predicateWithFormat:@"SELF MATCHES %@",expression];
    return [predicate evaluateWithObject:self];
}

-(BOOL) isTextFieldIntValue{
```

```

        return [self isTextFieldMatchWithRegularExpression:@"[-]{0,1}
[0-9]*"];
    }
    -(BOOL) isTextFieldUnsignedIntValue{
        return [self isTextFieldMatchWithRegularExpression:@"[0-9]+"];
    }

```

五. UITextField的键盘事件多次回调问题

1. 键盘高度遮挡问题

一般出现遮挡的时候我们用以下代码,看看当前textfield是否在键盘下面, 在的话算出键盘的顶端和textfield的底部的距离, 然后做偏移动画

```

- (void)keyboardWillShow:(NSNotification *)notification {

    NSDictionary *userInfo = [notification userInfo];

    NSValue* aValue = [userInfo
objectForKey:UIKeyboardFrameEndUserInfoKey];
    CGRect keyboardRect = [aValue CGRectValue];
    keyboardRect = [self.view convertRect:keyboardRect
fromView:nil];

    CGFloat keyboardTop = keyboardRect.origin.y;

    CGFloat offset = self.normalTextField.frame.size.height +
self.normalTextField.frame.origin.y - keyboardTop;

    NSValue *animationDurationValue = [userInfo
objectForKey:UIKeyboardAnimationDurationUserInfoKey];
    NSTimeInterval animationDuration;
    [animationDurationValue getValue:&animationDuration];

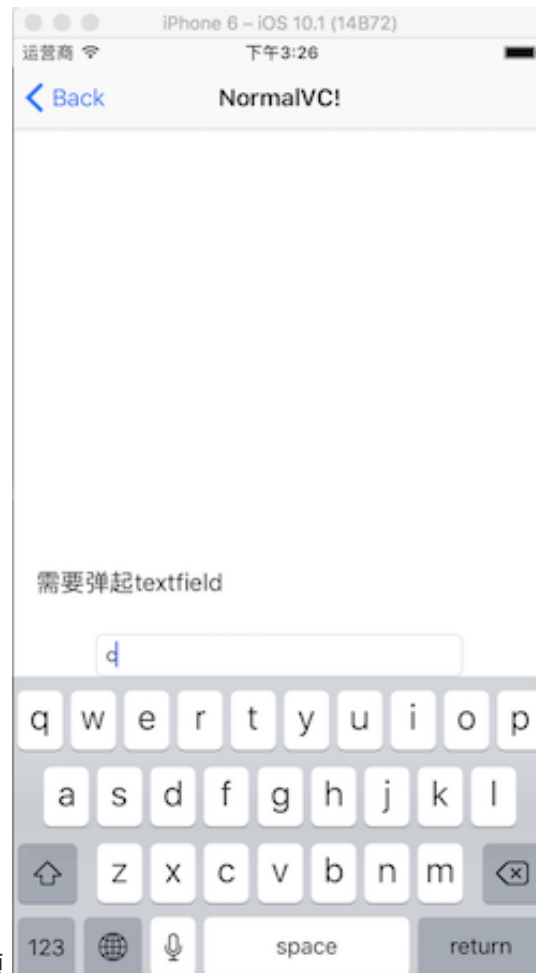
    if(offset > 0){
        // Animate the resize of the text view's frame in sync with
the keyboard's appearance.
        [UIView beginAnimations:nil context:NULL];
        [UIView setAnimationDuration:animationDuration];

        CGRect rect = CGRectMake(0.0f, -
offset,self.view.frame.size.width,self.view.frame.size.height);
        self.view.frame = rect;
        [UIView commitAnimations];
    }
}

```

```
}  
}
```

- 真机 如果使用了中文输入法，注册的keyboardWillShow会回调两次。第一次是键盘默认高度216，第二次则是加了keyboard的导航栏的高度。

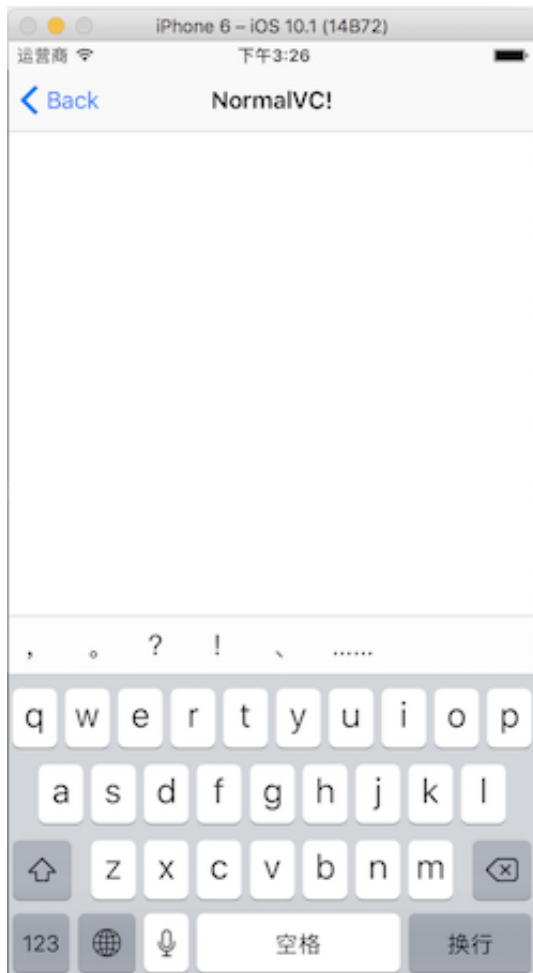


- 模拟器 第一次弹出键盘没有问题
userinfo: 打印

```
(lldb) po userinfo  
{  
    UIKeyboardAnimationCurveUserInfoKey = 7;  
    UIKeyboardAnimationDurationUserInfoKey = "0.25";  
    UIKeyboardBoundsUserInfoKey = "CGRect: {{0, 0}, {414, 226}}";  
    UIKeyboardCenterBeginUserInfoKey = "CGPoint: {207, 849}";  
    UIKeyboardCenterEndUserInfoKey = "CGPoint: {207, 623}";  
    UIKeyboardFrameBeginUserInfoKey = "CGRect: {{0, 736}, {414,  
226}}";  
    UIKeyboardFrameEndUserInfoKey = "CGRect: {{0, 510}, {414,  
226}}";  
}
```

```
UIKeyboardIsLocalUserInfoKey = 1;  
}
```

此时我们去按123旁边的小圆球会出现如下的图：



打印userinfo:

```
(lldb) po userInfo  
{  
    UIKeyboardAnimationCurveUserInfoKey = 7;  
    UIKeyboardAnimationDurationUserInfoKey = "0.25";  
    UIKeyboardBoundsUserInfoKey = "CGRect: {{0, 0}, {414, 271}}";  
    UIKeyboardCenterBeginUserInfoKey = "CGPoint: {207, 623}";  
    UIKeyboardCenterEndUserInfoKey = "CGPoint: {207, 600.5}";  
    UIKeyboardFrameBeginUserInfoKey = "CGRect: {{0, 510}, {414,  
226}}";  
    UIKeyboardFrameEndUserInfoKey = "CGRect: {{0, 465}, {414,  
271}}";  
    UIKeyboardIsLocalUserInfoKey = 1;  
}
```


键盘被遮挡了。

总结:观察结果,发现了这个规律,打印一下时间,还有一个问题就是,中文键盘第一次启动的时候会回调两次。

```
keyboardRect = [self.view convertRect:keyboardRect fromView:nil];
```

所以去掉这句话即可

六. 使用封装的XXTextField

UITextField, UITextView 中如果有keyboard的时候, 需要一个自动弹起事件, 以及弹起之后的content的偏移对父view的处理。如果每个页面都实现一次会非常复杂。这里我们介绍一种自动化的处理机制。在此之前, 先介绍一下文字处理框架.最后给大家推荐一下我写的 XXTextField ,大家也可以在此基础上自己添加一些正则表达式。使用方法很简单.欢迎加入QQ群: 237305299 , 一起探讨iOS技术问题

####1.解决uiview中的textfield 遮挡问题"

```
_textFieldName.keyboardType = UIKeyboardTypeDefault;
_textFieldName.inputType = XXTextFieldTypeOnlyInt;
_textFieldName.maxLength = 5;
_textfieldPwd.inputType = XXTextFieldTypeForbidEmoj;

#import "XXKeyboardManager.h"
@interface XXCorrectVC ()
<XXKeyboardManagerShowHiddenNotificationDelegate>
@end

@implementation XXCorrectVC

- (void)viewDidLoad {
    [super viewDidLoad];
    [[XXKeyboardManager sharedInstance] setDelegate:self];
    // Do any additional setup after loading the view from its nib.
}

#pragma mark- KeyBoardShow/Hidden
- (void)showKeyboardWithRect:(CGRect)keyboardRect
    withDuration:(CGFloat)animationDuration
{
    CGFloat offset = self.textFieldCorrect.frame.size.height +
self.textFieldCorrect.frame.origin.y - keyboardRect.origin.y;
```

```

        if(offset < 0){
            return;
        }
        [UIView animateWithDuration:animationDuration
                                delay:0.f
                                options:UIViewAnimationOptionCurveEaseInOut
                                animations:^(
                CGRect rect = CGRectMake(0.0f, -
offset, self.view.frame.size.width, self.view.frame.size.height);
                self.view.frame = rect;
            ) completion:^(BOOL finished) {

        }];
    }

- (void)hiddenKeyboardWithRect:(CGRect)keyboardRect
    withDuration:(CGFloat)animationDuration
{
    [UIView animateWithDuration:animationDuration
                            delay:0.f
                            options:UIViewAnimationOptionCurveEaseInOut
                            animations:^(
                self.textFieldCorrect.frame = self.view.bounds;
            ) completion:^(BOOL finished) {
        }];
}
@end

```

2.解决uitableview中键盘遮挡问题

```

/*
 * 键盘要显示的时候
 */
- (void)showKeyboardWithRect:(CGRect)keyboardRect
    withDuration:(CGFloat)animationDuration{

    CGSize kbSize = keyboardRect.size;

    UIEdgeInsets contentInsets = UIEdgeInsetsMake(0.0, 0.0,
kbSize.height, 0.0);
    _baseTableView.contentInset = contentInsets;
    _baseTableView.scrollIndicatorInsets = contentInsets;

    // If active text field is hidden by keyboard, scroll it so
it's visible
    // Your app might not need or want this behavior.

```

```
CGRect aRect = self.view.frame;
aRect.size.height -= kbSize.height;

if (!CGRectContainsPoint(aRect, _activeCell.frame.origin) ) {
    [_baseTableView scrollRectToVisible:_activeCell.frame
    animated:YES];
}

/*
 * 键盘要消失的时候
 */
- (void)hiddenKeyboardWithRect:(CGRect)keyboardRect
    withDuration:(CGFloat)animationDuration{
    _baseTableView.contentInset = UIEdgeInsetsZero;
    _baseTableView.scrollIndicatorInsets = UIEdgeInsetsZero;
}
```

下载地址:[xxtextfield](#)

七. 参考链接

1. [iOS中UITextField的字数限制](#)
2. [正则表达式语法表](#)
3. [Emoj过滤](#)
4. [UIKeyboardWillShowNotification调用多次的问题](#)
5. [苹果官方做法](#)