GeekBand 极客班

互联网人才 + 油站!

GeekBand 极客班

www.geekband.com

设计模式五

孔祥波

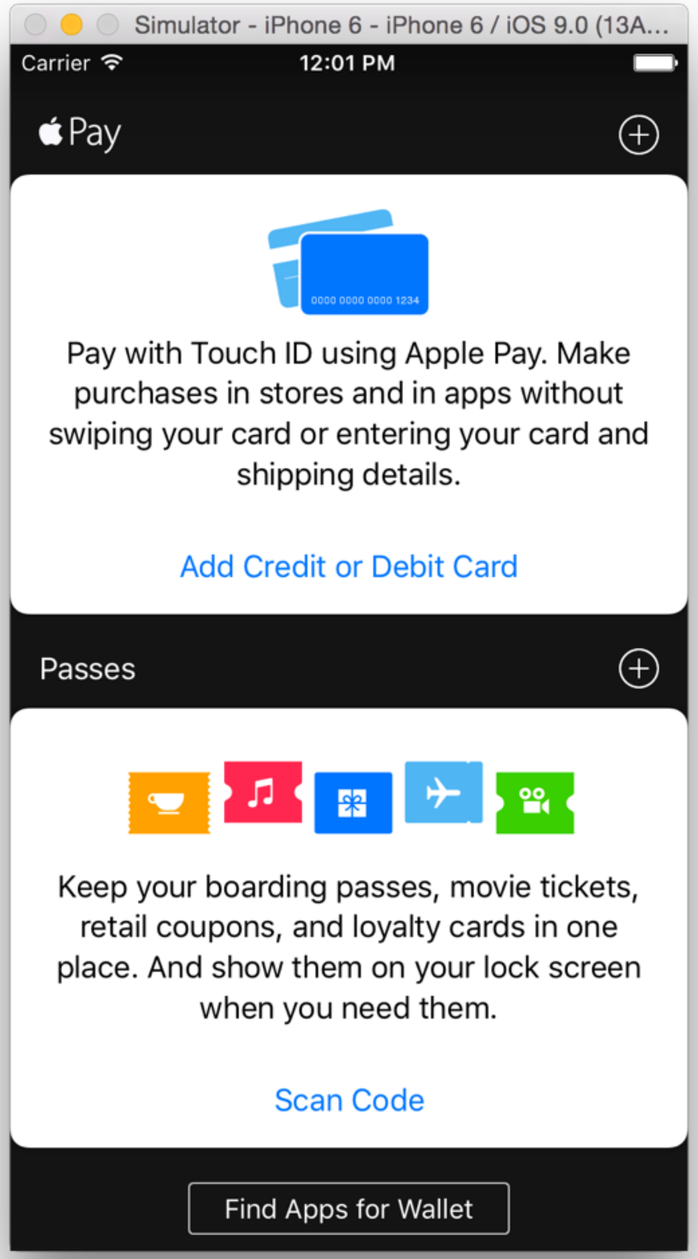
回顾

- 归档和解档(Serialization)
- 复制模式

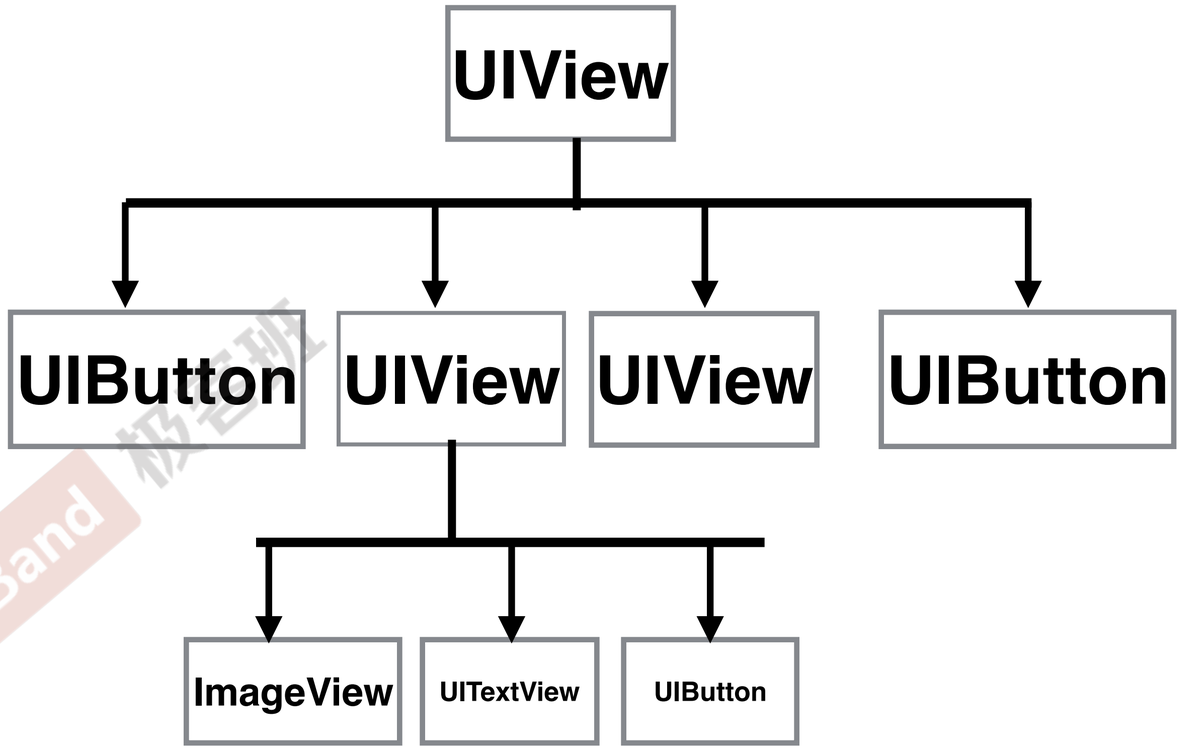
层次结构

动机

- 对象之间关系
- 允许一组相互协作的对象当成单一对象处理
- 无需子类化，实现自定义
- 降低parents class复杂度
- 使用tree 结构，方便数据的存储，操作，和搜索

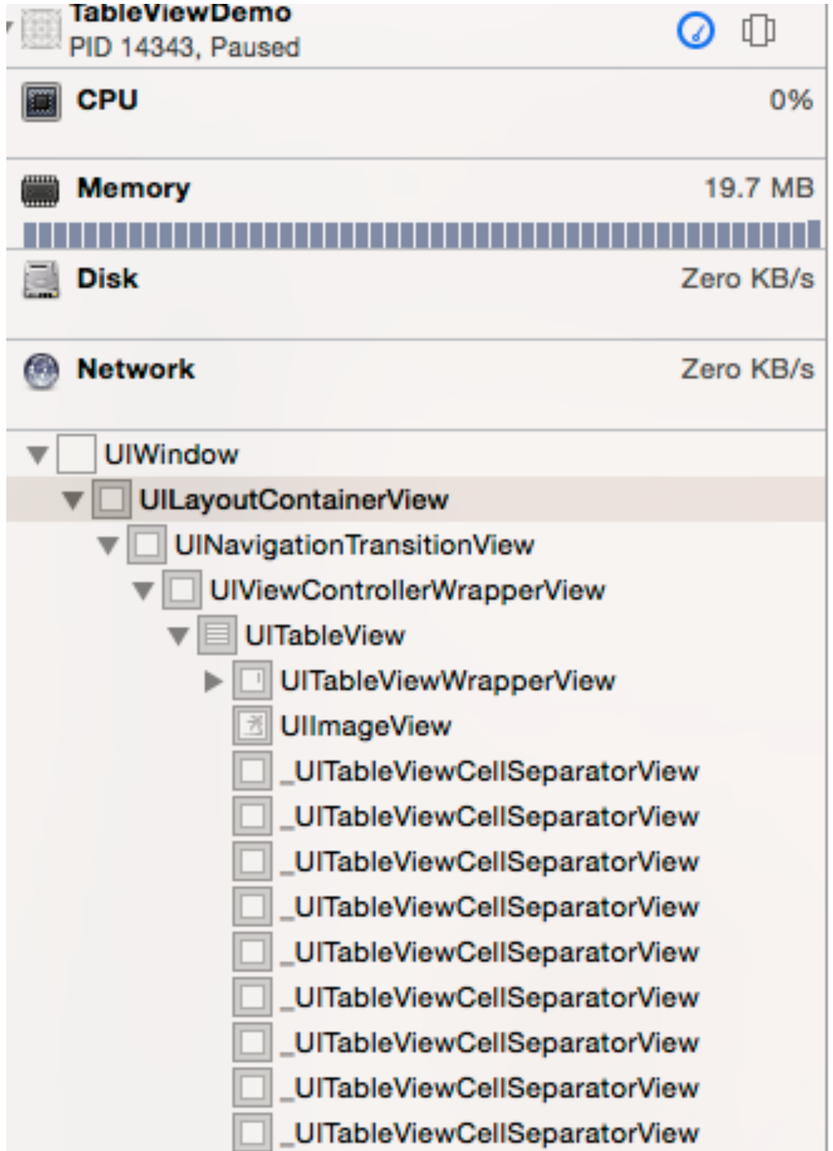


UIView Hierarchy



部分结构

UIView Hierarchy



View Hierarchy 操作

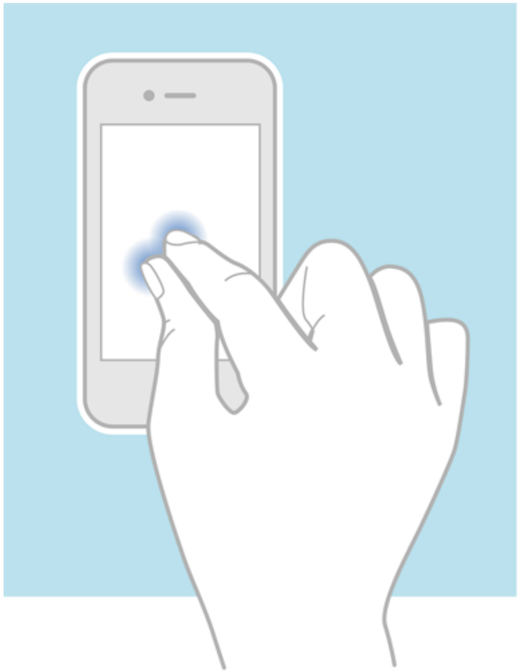
```
@interface UIView(UIViewHierarchy)

@property(nullable, nonatomic,readonly) UIView       *superview;
@property(nonatomic,readonly,copy) NSArray<__kindof UIView *> *subviews;
@property(nullable, nonatomic,readonly) UIWindow     *window;

- (void)removeFromSuperview;
- (void)insertSubview:(UIView *)view atIndex:(NSInteger)index;
- (void)exchangeSubviewAtIndex:(NSInteger)index1 withSubviewAtIndex:(NSInteger)index2;

- (void)addSubview:(UIView *)view;
- (void)insertSubview:(UIView *)view belowSubview:(UIView *)siblingSubview;
- (void)insertSubview:(UIView *)view aboveSubview:(UIView *)siblingSubview;

- (void)bringSubviewToFront:(UIView *)view;
- (void)sendSubviewToBack:(UIView *)view;

- (void)didAddSubview:(UIView *)subview;
- (void)willRemoveSubview:(UIView *)subview;

- (void)willMoveToSuperview:(nullable UIView *)newSuperview;
- (void)didMoveToSuperview;
- (void)willMoveToWindow:(nullable UIWindow *)newWindow;
- (void)didMoveToWindow;

- (BOOL)isDescendantOfView:(UIView *)view;  // returns YES for self.
- (nullable UIView *)viewWithTag:(NSInteger)tag;     // recursive search. includes self

// Allows you to perform layout before the drawing cycle happens. -layoutIfNeeded forces layout early
- (void)setNeedsLayout;
- (void)layoutIfNeeded;
```

iOS rendering tree

- UIView 负责界面显示和事件处理
- CALayer 负责屏幕渲染(Layer Tree)
- View/Layer 的变化需要通过渲染器实时渲染到屏幕上
- layer.presentationLayer

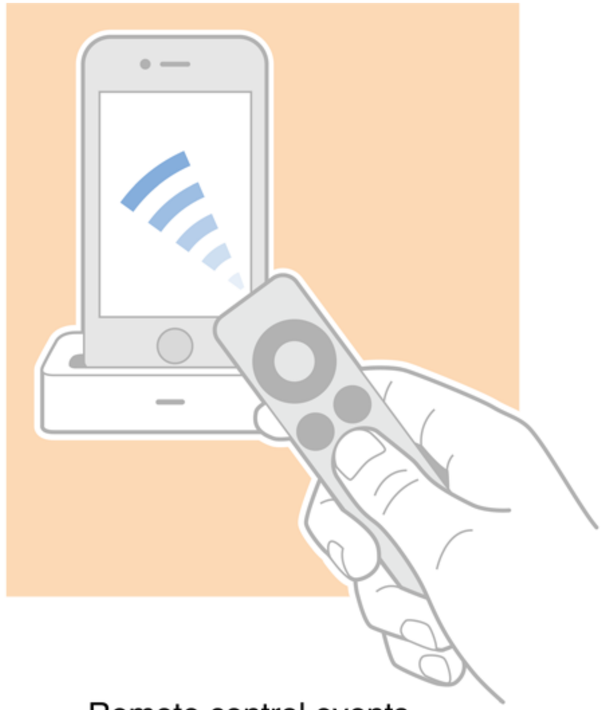
响应链(Responder chain)

Event on iOS



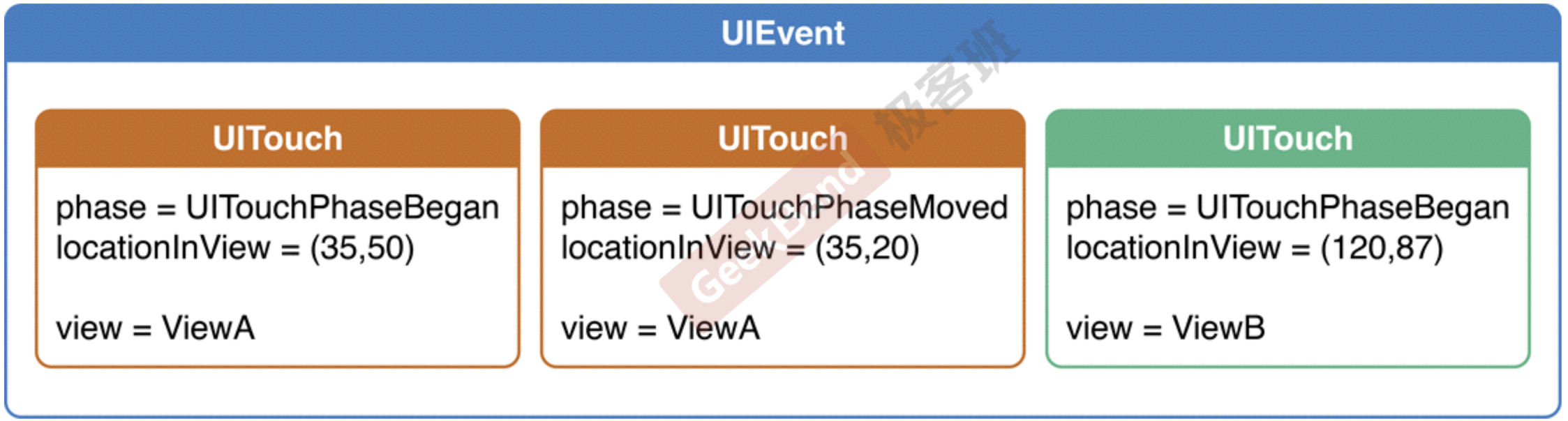
Multitouch events

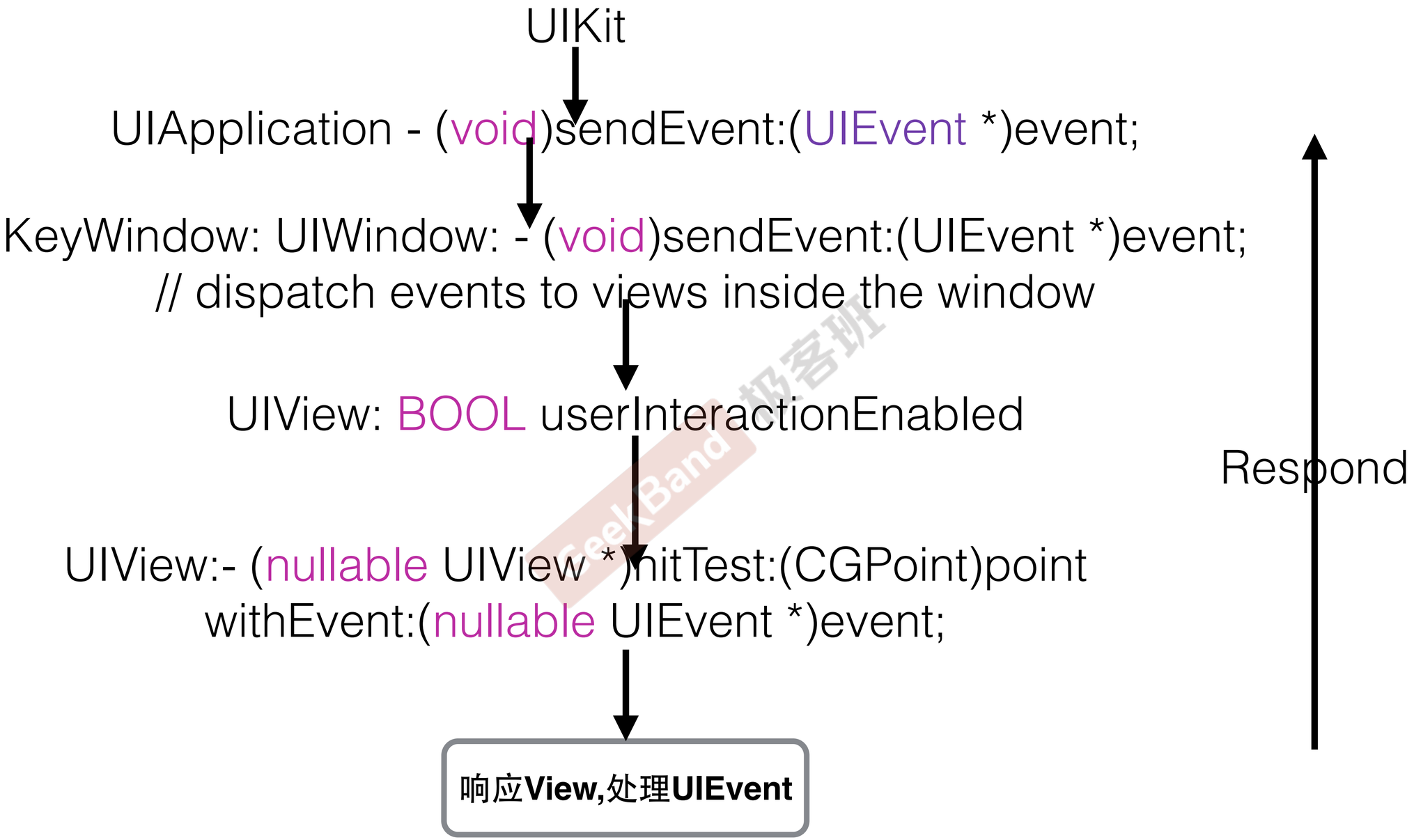
Accelerometer events



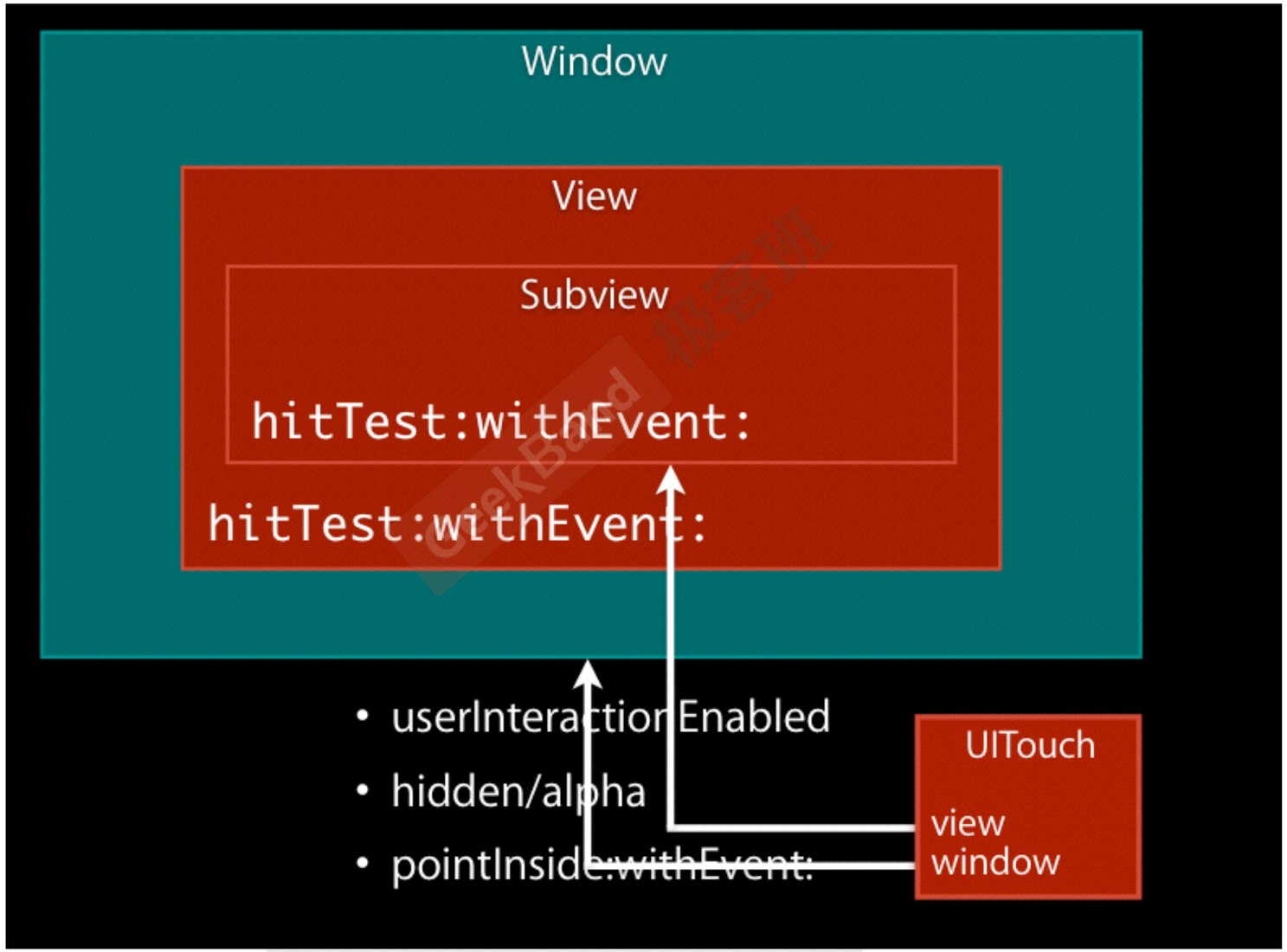
Remote control events

UIEvent概念

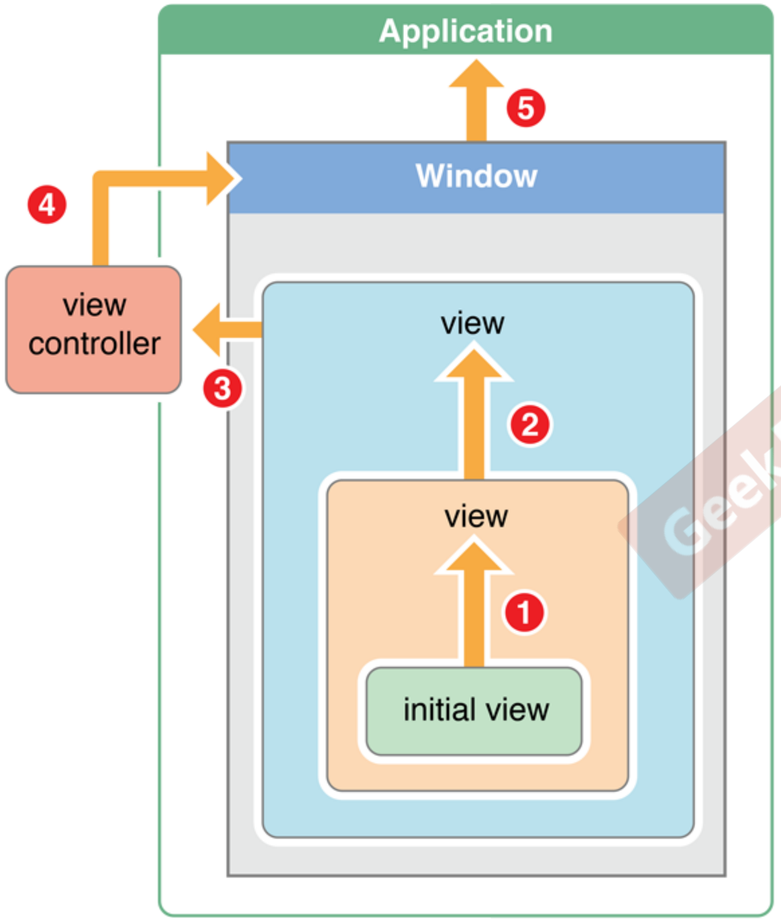

UIEvent
UITouch
phase = UITouchPhaseBegan
locationInView = (35,50)
view = ViewA
UITouch
phase = UITouchPhaseMoved
locationInView = (35,20)
view = ViewA
UITouch
phase = UITouchPhaseBegan
locationInView = (120,87)
view = ViewB

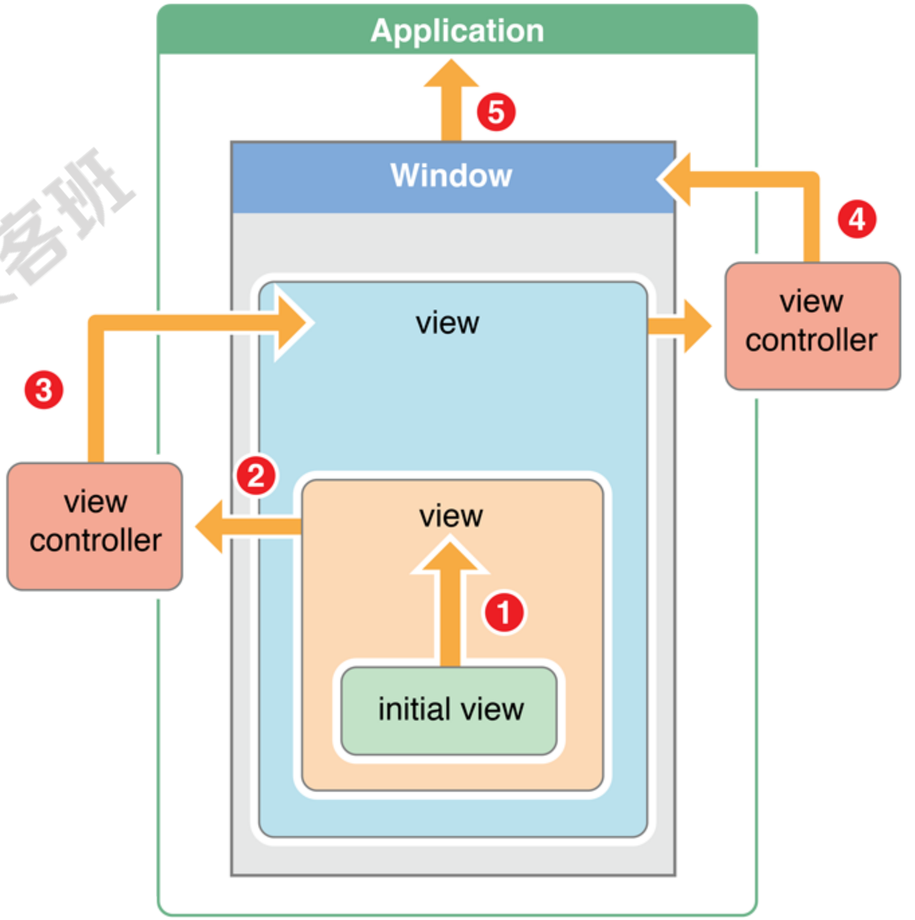

UIKit
UIApplication - (void)sendEvent:(UIEvent *)event;
KeyWindow: UIWindow: - (void)sendEvent:(UIEvent *)event;
// dispatch events to views inside the window
UIView: BOOL userInteractionEnabled
UIView:- (nullable UIView *)hitTest:(CGPoint)point
withEvent:(nullable UIEvent *)event;
响应View,处理UIEvent
Respond

Hit-Testing Returns the View Where a Touch Occurred



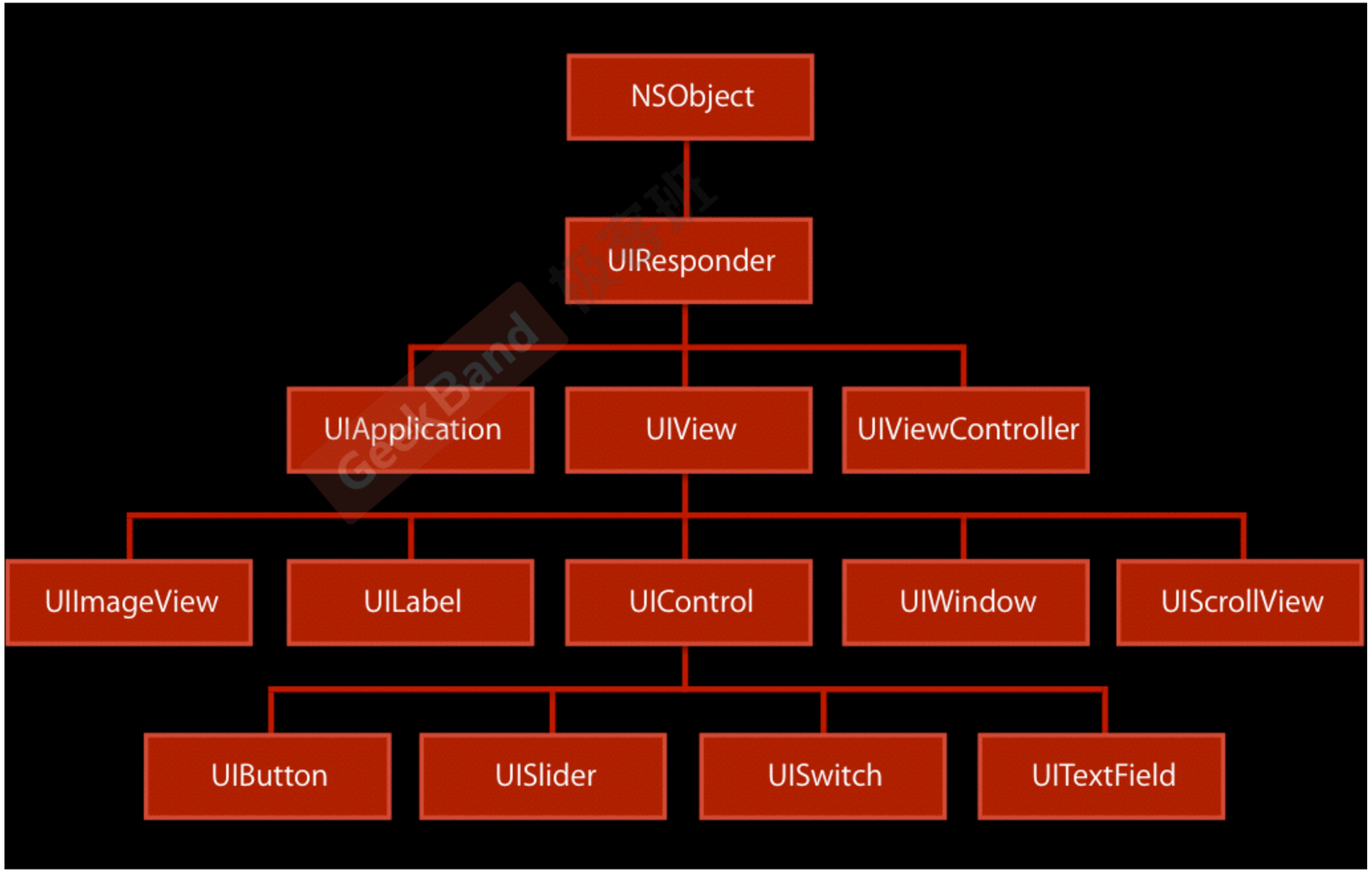
响应链分发路径





```
NS_CLASS_AVAILABLE_IOS(2_0) @interface UIResponder : NSObject

- (nullable UIResponder*)nextResponder;

- (BOOL)canBecomeFirstResponder;    // default is NO
- (BOOL)becomeFirstResponder;

- (BOOL)canResignFirstResponder;    // default is YES
- (BOOL)resignFirstResponder;

- (BOOL)isFirstResponder;

// Generally, all responders which do custom touch handling should override all four of these methods.
// Your responder will receive either touchesEnded:withEvent: or touchesCancelled:withEvent: for each
// touch it is handling (those touches it received in touchesBegan:withEvent:).
// *** You must handle cancelled touches to ensure correct behavior in your application.  Failure to
// do so is very likely to lead to incorrect behavior or crashes.
- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(nullable UIEvent *)event;
- (void)touchesMoved:(NSSet<UITouch *> *)touches withEvent:(nullable UIEvent *)event;
- (void)touchesEnded:(NSSet<UITouch *> *)touches withEvent:(nullable UIEvent *)event;
- (void)touchesCancelled:(nullable NSSet<UITouch *> *)touches withEvent:(nullable UIEvent *)event;

- (void)motionBegan:(UIEventSubtype)motion withEvent:(nullable UIEvent *)event NS_AVAILABLE_IOS(3_0);
- (void)motionEnded:(UIEventSubtype)motion withEvent:(nullable UIEvent *)event NS_AVAILABLE_IOS(3_0);
- (void)motionCancelled:(UIEventSubtype)motion withEvent:(nullable UIEvent *)event NS_AVAILABLE_IOS(3_0);

- (void)remoteControlReceivedWithEvent:(nullable UIEvent *)event NS_AVAILABLE_IOS(4_0);

- (BOOL)canPerformAction:(SEL)action withSender:(nullable id)sender NS_AVAILABLE_IOS(3_0);
// Allows an action to be forwarded to another target. By default checks -canPerformAction:withSender: to either
    self, or go up the responder chain.
- (nullable id)targetForAction:(SEL)action withSender:(nullable id)sender NS_AVAILABLE_IOS(7_0);
```

UIKit 继承关系


NSObject
UIResponder
UIApplication
UIView
UIViewController
UIImageView
UILabel
UIControl
UIWindow
UIScrollView
UIButton
UISlider
UISwitch
UITextField

小结

- 层次结构
- 响应链(Responder chain)