



# NotificationCenter

# NotificationCenter

- broadcast mechanism for notifications
- subscribe and receive messages
- one-to-many



# What is notification?

System-wide events like

`Notification.Name.UIKeyboardWillShow` // for handling showing keyboard

API-specific events like

`UserDefaults.didChangeNotification` // for reacting on changing  
`UserDefaults`

Create your own

```
extension Notification.Name {  
    static let didAddAccount = Notification.Name("Account added")  
    static let didRemoveAccount = Notification.Name("Account removed")  
}
```

# Subscribe and receive

```
NotificationCenter.default.addObserver(self, selector:
#selector(handle), name: .UIKeyboardWillShow, object: nil)

func handle(notification: Notification) {

}
```

Don't forget to unsubscribe:

```
NotificationCenter.default.removeObserver(self) // for every
NotificationCenter.default.removeObserver(self,
name: .UIKeyboardWillShow, object: nil) // for specific
```

## Post notification

```
NotificationCenter.default.post(name: .didAddAccount, object:
nil)
```

# Notification explained

```
// obviously the name of notification
var name: Notification.Name

// value to attach to notification(can be filtered by on
subscription stage)
var object: Any?

// dictionary with some additional information
var userInfo: [AnyHashable : Any]?
```