

Figure 3-14. Remove the automatically built width constraint at build time so that we can replace it in code

Now create an outlet for this red view in your code (I’ve named mine “v”) and add the following code to your viewDidLoad:

```
v.widthAnchor.constraintEqualToAnchor(btn2.widthAnchor,
    constant:0).active = true
```

## See Also

## 3.4 Allowing Users to Enter Text in Response to Local and Remote Notifications

### Problem

You want to allow your users to enter some text in response to local or push notifications that you display. And you would additionally like to be able to read this text in your app and take action on it.

## Solution

Set the new behavior property of the `UIUserNotificationAction` class to `.TextInput` (with a leading period).

## Discussion

Let's say that we want our app to register for local notifications and then ask the user for her name once the app has been sent to the background. The user enters her name and then we come to the foreground and take action on that name.

We start by writing a method that allows us to register for local notifications:

```
func registerForNotifications(){

    let enterInfo = UIMutableUserNotificationAction()
    enterInfo.identifier = "enter"
    enterInfo.title = "Enter your name"
    enterInfo.behavior = .TextInput //this is the key to this example
    enterInfo.activationMode = .Foreground

    let cancel = UIMutableUserNotificationAction()
    cancel.identifier = "cancel"
    cancel.title = "Cancel"

    let category = UIMutableUserNotificationCategory()
    category.identifier = "texted"
    category.setActions([enterInfo, cancel], forContext: .Default)

    let settings = UIUserNotificationSettings(
        forTypes: .Alert, categories: [category])

    UIApplication.sharedApplication()
        .registerUserNotificationSettings(settings)
}
```

We set the behavior property on the `UIMutableUserNotificationAction` instance to `.TextInput` to allow this particular action to receive text input from the user. Now we will move on to calling this method when our app is launched:

```
func application(application: UIApplication,
    didFinishLaunchingWithOptions
    launchOptions: [NSObject : AnyObject]?) -> Bool {

    registerForNotifications()
```

```

        return true
    }

```

We also need a method to schedule a local notification whenever asked for:

```

func scheduleNotification(){

    let n = UILocalNotification()
    let c = NSCalendar.autoupdatingCurrentCalendar()
    let comp = c.componentsInTimeZone(c.timeZone, fromDate: NSDate())
    comp.second += 3
    let date = c.dateFromComponents(comp)
    n.fireDate = date

    n.alertBody = "Please enter your name now"
    n.alertAction = "Enter"
    n.category = "texted"
    UIApplication.sharedApplication().scheduleLocalNotification(n)

}

```

And I'll call this method when our app is sent to the background:

```

func applicationDidEnterBackground(application: UIApplication) {
    scheduleNotification()
}

```

Once that is done, we will read the text that the user has entered and do our work with it (I'll leave this to you):

```

func application(application: UIApplication,
    handleActionWithIdentifier identifier: String?,
    forLocalNotification notification: UILocalNotification,
    withResponseInfo responseInfo: [NSObject : AnyObject],
    completionHandler: () -> Void) {

    if let text = responseInfo[UIUserNotificationActionResponseTypedTextKey]
        as? String{

        print(text)
        //TODO: now you have access to this text

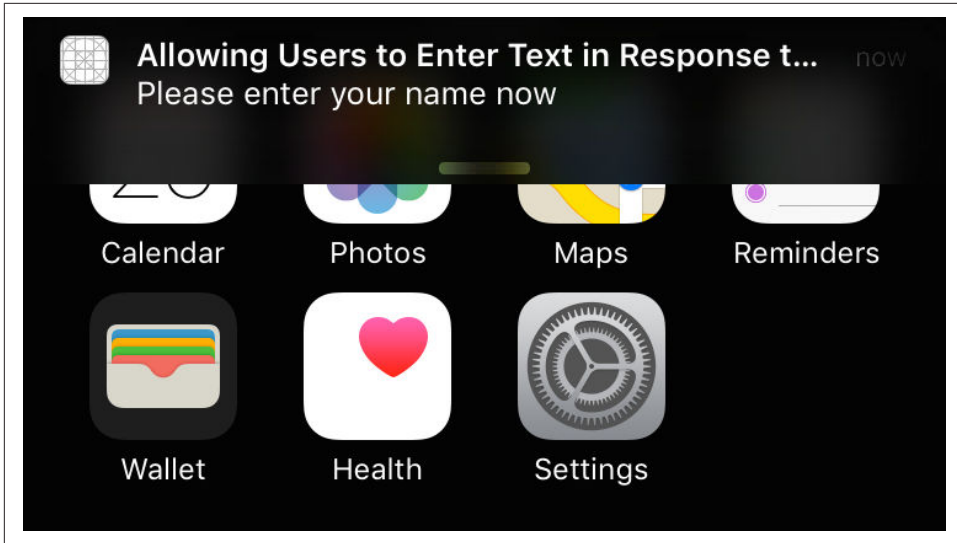
    }

    completionHandler()
}

```

}

Let's run it and then send the app to the background and see what happens (see [Figure 3-15](#)).



*Figure 3-15. A local notification is shown on the screen*

Then take that little bar at the bottom of the notification and drag it down to show the actions that are possible on the notification (see [Figure 3-16](#)).

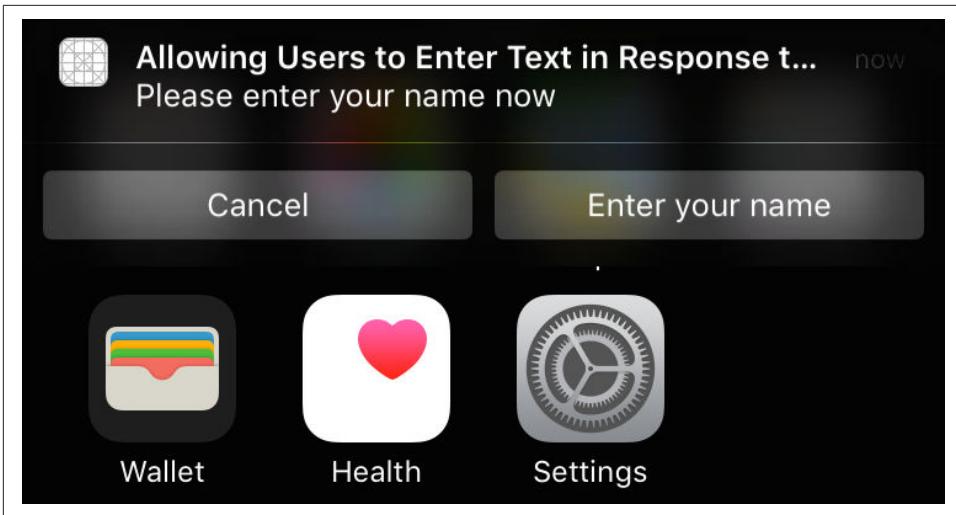


Figure 3-16. Possible actions on our local notification

Now if the user just taps on the Enter button, she will see a text field and can then enter her information. Upon submitting the text, she will be redirected to our app where we will receive the text (see [Figure 3-17](#)).

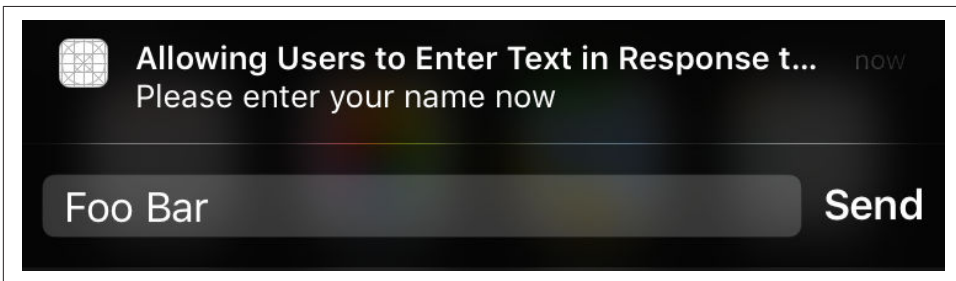


Figure 3-17. Entering text in a local notification

## See Also

## 3.5 Dealing with Stacked Views in Code

### Problem

You want to programmatically manipulate the contents of stack views.

### Solution

Use an instance of the `UIStackView`.