

探究响应式编程 在 iOS 开发中的优 势

iDev
前青 K

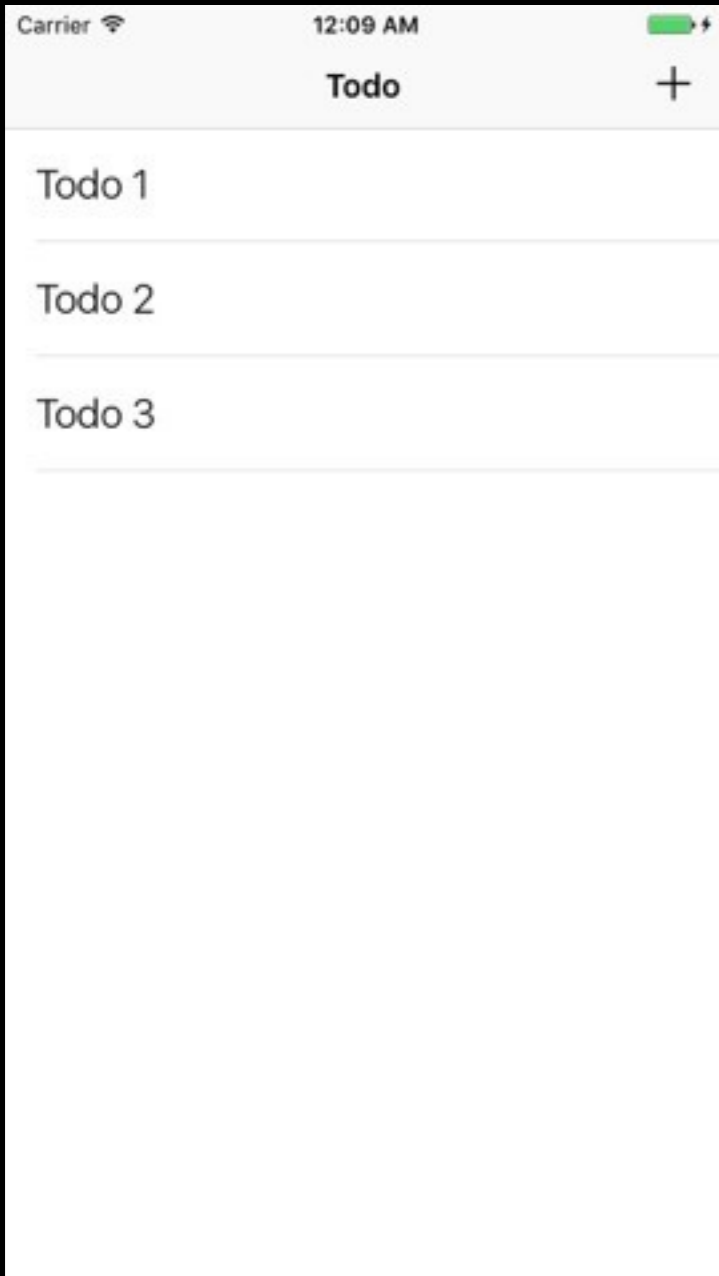


•

•







COMPLETAR (TODO 4 , ac. 5)

Todo 4

LOGOUT (1000 4 , ac. 0)

手动 维护视图

```
tableView.insert( todo 4 , at: 0 )
```

变动

增加 **Todo 4**



```
tableView.beginUpdates()  
tableView.insertRows(at: [IndexPath(row: 0, section: 0)],  
                    with: .automatic)  
tableView.endUpdates()
```


视图

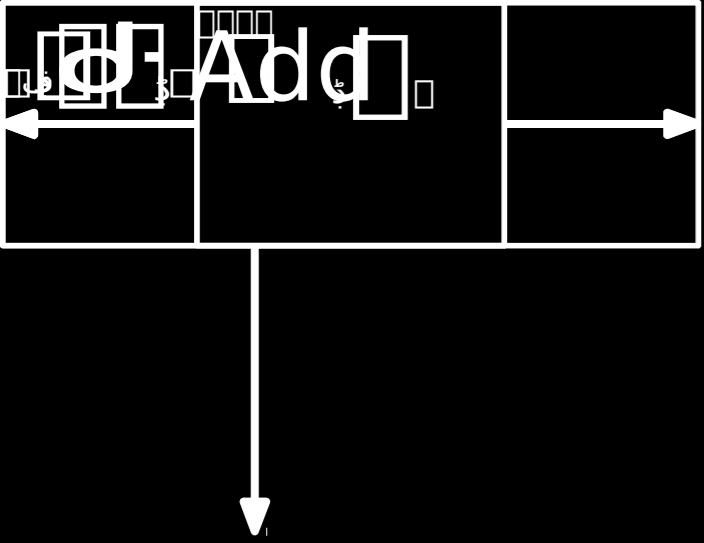
数据

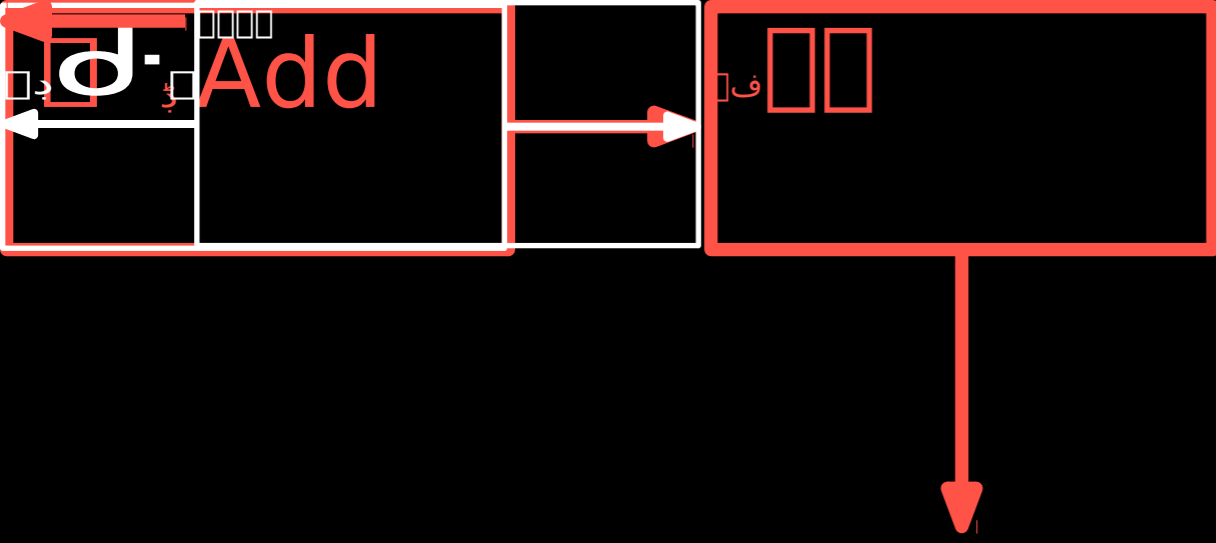
我 (todoList) 有变动

我在顶部增加了“**Todo 4**”

通知

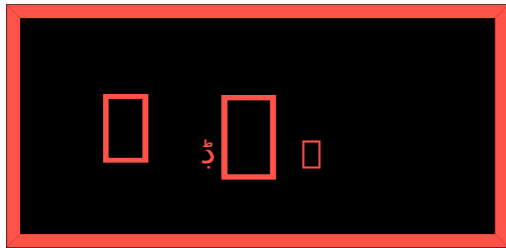
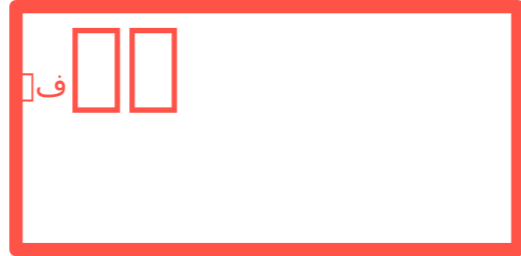
订阅

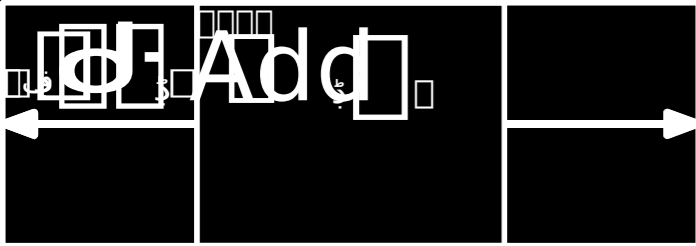




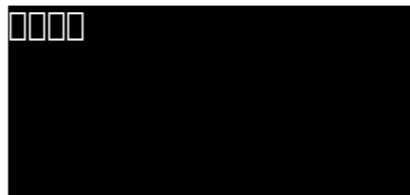


.





数据



no,anQK

Expandable
)
■ o;
anOK

E
8
2

Run, ng h pandable

响 wController.swat) 囚由 tew肺, Sel 刷刷
ω

Z
e
r
o

K
B
/
s

u
e

,
c
o
m
.
.
.
t
v
e
a
d

<
s
e
r
i
a

)
e
d
-
f
a
t
o
l
E
r
r

O
r
M
e
s
s
a
g
e
.
.

始
w

s
e
e
e
t
R
o
w
A
t
l
n
d
.
.
叭
;
e
w

-
U
S
8
'
S
e
l
e
c

1
W
:
:
它
A
C
o
m
m
O
e
l
e
r
r
回
B
I
:
:
:
A
t
t
e
r
C
A
F
I
u
s
h
A
n
d
民
u
n
:
:
0
m
m
,
协
<
a
n
d
l
e
r

,
日
P
-
I
S
-
C
A
L
L
I
N
G
"
日
1
c
a
t
,
o
n
阳
,
1
)
:
a
1
1
n
M
a
,
n

```
164 func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
165     let indexofTappedRow = visibleRowsPerSection[indexPath.section][indexPath.row]
166
167     if cellDescriptors[indexPath.section][indexofTappedRow]["isExpandable"] as! Bool == true {
168         var shouldExpandAndShowSubRows = false
169         if cellDescriptors[indexPath.section][indexofTappedRow]["isExpanded"] as! Bool == false {
170             // In this case the cell should expand.
171             shouldExpandAndShowSubRows = true
172         }
173
174         cellDescriptors[indexPath.section][indexofTappedRow]["isExpanded"] = shouldExpandAndShowSubRows
175
176         for i in (indexofTappedRow + 1)...(indexofTappedRow + (cellDescriptors[indexPath.section][indexofTappedRow]
177             ["additionalRows"] as! Int)) {
178             cellDescriptors[indexPath.section][i]["isVisible"] = shouldExpandAndShowSubRows
179         }
180     } else {
181         if cellDescriptors[indexPath.section][indexofTappedRow]["cellIdentifier"] as! String == "idCellValuePicke
182             var indexofParentCell: Int!
183
184         for i in stride(from: indexofTappedRow - 1, to: 0, by: -1) {
185             if cellDescriptors[indexPath.section][i]["isExpandable"] as! Bool == true {
186                 indexofParentCell = i
187                 break
188             }
189         }
190
191         cellDescriptors[indexPath.section][indexofParentCell]["cellIdentifier"] = (tblExpandable.cellIdentifier(at:
192             indexPath.section, row: indexofParentCell) as! CustomCell).textLabel?.text
193         cellDescriptors[indexPath.section][indexofParentCell]["isExpanded"] = true
194         for i in (indexofParentCell + 1)...(indexofParentCell + (cellDescriptors[indexPath.section][indexofPa
195             ["additionalRows"] as! Int)) {
196             cellDescriptors[indexPath.section][i]["isVisible"] = true
197         }
198     }
199
200     getIndicesOfVisibleRows()
201     tblExpandable.reloadSections(IndexSet(integer: indexPath.section), with: UITableViewRowAnimation.fade)
202 }
```



Thread 1: EXC_BREAKPOINT (code=1, su

indexPath.section][indexofPa


```
// 任务一
for var i=indexOfTappedRow - 1; i>=0; --i {
    if cellDescriptors[indexPath.section][i]["isExpandable"] as! Bool == true {
        indexOfParentCell = i
        break
    }
}
// 任务二
```



```
tblExpandable.register(UINib(nibName: "NormalCell", bundle: nil), forCellReuseIdentifier: "idCellNormal")
tblExpandable.register(UINib(nibName: "TextFieldCell", bundle: nil), forCellReuseIdentifier: "idCellTextField")
tblExpandable.register(UINib(nibName: "DatePickerCell", bundle: nil), forCellReuseIdentifier: "idCellDatePicker")
tblExpandable.register(UINib(nibName: "SwitchCell", bundle: nil), forCellReuseIdentifier: "idCellSwitch")
tblExpandable.register(UINib(nibName: "ValuePickerCell", bundle: nil), forCellReuseIdentifier: "idCellValuePicker")
tblExpandable.register(UINib(nibName: "SliderCell", bundle: nil), forCellReuseIdentifier: "idCellSlider")
```

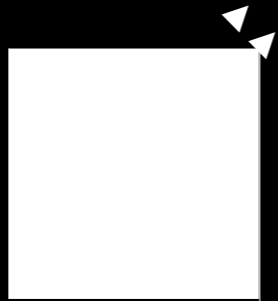


```
if cellDescriptors[indexPath.section][indexOfTappedRow]["isExpandable"] as! Bool == true {
    var shouldExpandAndShowSubRows = false
    if cellDescriptors[indexPath.section][indexOfTappedRow]["isExpanded"] as! Bool == false {
        // In this case the cell should expand.
        shouldExpandAndShowSubRows = true
    }

    cellDescriptors[indexPath.section][indexOfTappedRow]["isExpanded"] = shouldExpandAndShowSubRows

    for i in (indexOfTappedRow + 1)..(indexOfTappedRow + (cellDescriptors[indexPath.section][indexOfTappedRow]
        ["additionalRows"] as! Int)) {
        cellDescriptors[indexPath.section][i]["isVisible"] = shouldExpandAndShowSubRows
    }
} else {
    if cellDescriptors[indexPath.section][indexOfTappedRow]["cellIdentifier"] as! String == "idCellValuePicker" {
        var indexOfParentCell: Int!

        for i in stride(from: indexOfTappedRow - 1, to: -1, by: -1) {
            if cellDescriptors[indexPath.section][i]["isExpandable"] as! Bool == true {
                indexOfParentCell = i
                break
            }
        }
    }
}
```

השם המלא של המערכת הוא: **מחשב אישי**


```
(title => textLabel.rx.text)
```

```
viewModel.title = textLabel.text
```

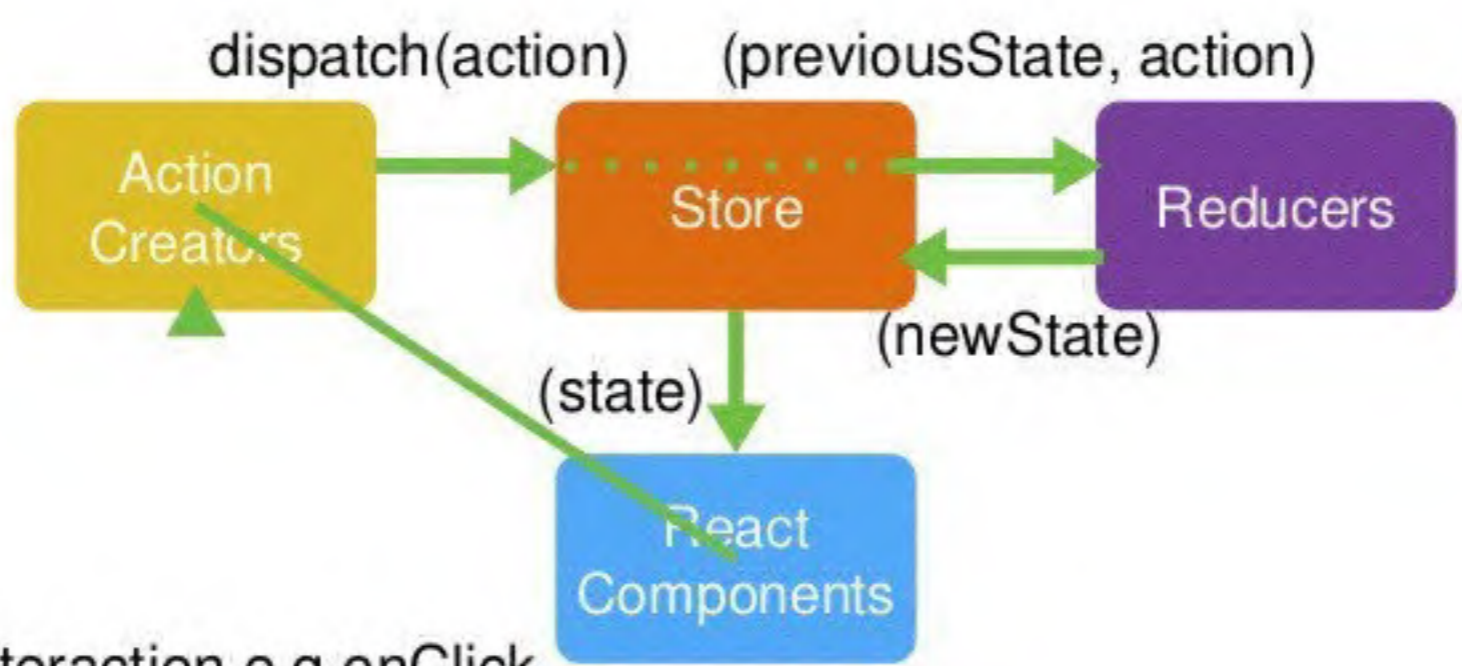
```
viewModel.title = textLabel.text
```

```
(datePickerCell.rx.date <-> date)
```

```
(datePickerCell.rx.date <-> date)
```

```
(switchCell.rx.isOn <-> isOn)
```


Redux Data Flow



Interaction e.g `onClick`

