Select multiple rows in tableview and tick the selected ones

Asked 8 years, 5 months ago Modified 4 years, 1 month ago Viewed 67k times

34

I'm loading a tableView from a plist file. This works with no problems. I just simply want to "tick" the selected rows. At the moment, with my code it didn't work as desired. At the moment, it looks as below:

- tap row1 (it will tick row 1 = good)
- tap row1 again (nothing happens = bad. I expect here the row to be unticked) While tapping again on row 1, it unticks then. After the second tap on it.
- when I tap row0 at the initial load of the tableview it never ticks me the row

my code:

```
class portals: UITableViewController {

    var lastSelectedIndexPath = NSIndexPath(forRow: -1, inSection: 0)

...

    override func tableView(tableView: UITableView, cellForRowAtIndexPath
indexPath: NSIndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCellWithIdentifier("myCell",
forIndexPath: indexPath) as! UITableViewCell

        // Configure the cell...
        cell.textLabel!.text = portals[indexPath.row]

        return cell
    }


    // Check which portal is selected
    override func tableView(tableView: UITableView, didSelectRowAtIndexPath
indexPath: NSIndexPath) {

        var whichPortalIsSelected: String = ""

        // Get Cell Label
        let indexPath = tableView.indexPathForSelectedRow();

        // Tick the selected row
        if indexPath!.row != lastSelectedIndexPath?.row {

            let newCell = tableView.cellForRowAtIndexPath(indexPath!)
            newCell?.accessoryType = .Checkmark

            lastSelectedIndexPath = indexPath

            whichPortalIsSelected = newCell!.textLabel!.text!
            println("You selected cell #\(lastSelectedIndexPath.row)!") //PPP
            println("You selected portal #\(whichPortalIsSelected)!") //PPP
        // Un-Tick unselected row
        } else {
            let newCell = tableView.cellForRowAtIndexPath(indexPath!)
            newCell?.accessoryType = .None

            whichPortalIsSelected = newCell!.textLabel!.text!
            println("You unselected cell #\(indexPath!.row)!") //PPP
            println("You unselected portal #\(whichPortalIsSelected)!") //PPP
        }

    }
}
```

swift

Share Follow

edited Nov 13, 2018 at 11:59
Neeraj Shukla
295 3 9

asked May 28, 2015 at 20:25
tefter
341 1 3 6

5 Answers

Sorted by: Highest score (default)

91

Swift 4

First, make your tableView support multiple selection :

```
self.tableView.allowsMultipleSelection = true
self.tableView.allowsMultipleSelectionDuringEditing = true
```

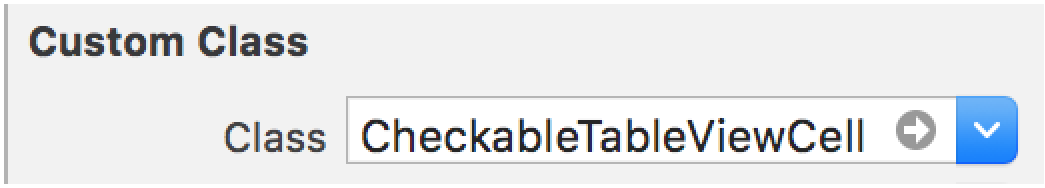
Then simply subclass UITableViewCell like this :

```
class CheckableTableViewCell: UITableViewCell {
    override init(style: UITableViewCellStyle, reuseIdentifier: String?) {
        super.init(style: style, reuseIdentifier: reuseIdentifier)
        self.selectionStyle = .none
    }

    required init?(coder aDecoder: NSCoder) {
        super.init(coder: aDecoder)
    }

    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)
        self.accessoryType = selected ? .checkmark : .none
    }
}
```

Finally, use it in your `cellForRowAt indexPath` as such :

```
let cell = tableView.dequeueReusableCell(withIdentifier: "cell",
    for: indexPath) as? CheckableTableViewCell
```

If you have to, don't forget to **subclass your prototype cell** in your xib/storyboard :



Custom Class

Class CheckableTableViewCell

Share Follow

edited Oct 1, 2019 at 12:07

answered Jan 27, 2016 at 14:45

Alexandre G.

1,779 14 17

This Way is very useful please tell me how can I get the index of the selected cells in table view or better to guide me if I have array of more informations about cells how can I have an array of information of the selected cells not all of them – Saeed Rahmatolahi Jul 16, 2017 at 11:28

Unfortunately, when scrolling through a tableView, the `isSelected` property of a selected cell seems to get lost when using reusable table view cells. So you can select multiple cells, get the checkmark, scroll down in the tableView, scroll up again and the cells will be unchecked again. Any ideas? – Alienbash Mar 5, 2018 at 16:47

You're right @Alienbash, I modified my answer accordingly. – Alexandre G. Mar 5, 2018 at 20:30

This answer is not valid without @Aatish Rajkarnikar answer, where you need to set you tableView from Selection to Multiple Selection. – nja Sep 26, 2018 at 7:23

Thanks for the heads up @nja – Alexandre G. Sep 26, 2018 at 14:09

15

First of all, go to your Storyboard and select you tableview and in the Attributes Inspector, set **Selection** to **Multiple Selection**.

Attributes Inspector with multiple selection

Then, override the setSelected(_ selected: Bool, animated: Bool) function in the subclass of UITableViewCell.

```
override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)
        accessoryType = selected ? UITableViewCellAccessoryType.checkmark :
UITableViewCellAccessoryType.none
    }
```

Share Follow

answered Jan 24, 2017 at 10:12

Aatish Rajkarnikar

201 3 4

1 Isn't this the best or right way to set the selected mark rather than cellForRowAt and didSelectRowAt? – Tibin Thomas Jan 9, 2019 at 5:59

11

This enable untick.

```
class TableViewController: UITableViewController
{
    var lastSelectedIndexPath = NSIndexPath(forRow: -1, inSection: 0)

    override func tableView(tableView: UITableView, cellForRowAtIndexPath
indexPath: NSIndexPath) -> UITableViewCell
    {
        let cell = tableView.dequeueReusableCellWithIdentifier("myCell",
forIndexPath: indexPath)

        // Configure the cell...
        cell.textLabel!.text = "row: \(indexPath.row)"

        if cell.selected
        {
            cell.selected = false
            if cell.accessoryType == UITableViewCellAccessoryType.None
            {
                cell.accessoryType = UITableViewCellAccessoryType.Checkmark
            }
            else
            {
                cell.accessoryType = UITableViewCellAccessoryType.None
            }
        }

        return cell
    }

    override func tableView(tableView: UITableView, didSelectRowAtIndexPath
indexPath: NSIndexPath)
    {
        let cell = tableView.cellForRowAtIndexPath(indexPath)

        if cell!.selected
        {
            cell!.selected = false
            if cell!.accessoryType == UITableViewCellAccessoryType.None
            {
                cell!.accessoryType = UITableViewCellAccessoryType.Checkmark
            }
            else
            {
                cell!.accessoryType = UITableViewCellAccessoryType.None
            }
        }
    }

    override func tableView(tableView: UITableView, numberOfRowsInSection
section: Int) -> Int
    {
        return 100
        }
    }
```

Share Follow

edited Nov 4, 2015 at 9:08
MichaelS
5,976 6 31 46

answered Nov 4, 2015 at 8:45
Liew Tze Hau
113 1 6

This is perfect. Thanks. – Craig.Pearce May 4, 2016 at 3:37

5

you have to make a costume class to get the selected state of the cell where you must override a func called setSelected(_ selected: Bool, animated: Bool) or the tick will be displayed randomly as you scroll ... here is an example of what i did: 1- created a class for the cell 2- added an outlet for an image to display the tick (you can escape this if you don't want a costume tick image) 3- overrided the function and used the param selected :D

here is my class:

```
import UIKit

class AddLocationCell: UITableViewCell {

     @IBOutlet weak var check: UIImageView!

        override func awakeFromNib() {
            super.awakeFromNib()
            // Initialization code
        }

        override func setSelected(_ selected: Bool, animated: Bool) {
            super.setSelected(selected, animated: animated)
            if selected{
                check.image = UIImage(named:"check_active")
            }else{
                check.image = UIImage(named:"check_normal")

            }
            // Configure the view for the selected state
        }
}
```

Share Follow

edited Aug 17, 2018 at 13:48
nishee
141 7

answered Oct 24, 2016 at 10:53
mohammad alnajjar
61 1 1

3

There are many solutions to this problem, here's one I came up with. I am using the built in cell "selected" property so the tableview saves it for us. Just make sure in your storyboard or when you setup the tableview in code you use multiple selection.

```
import UIKit

class TableViewController: UITableViewController
{
    var lastSelectedIndexPath = NSIndexPath(forRow: -1, inSection: 0)

    override func tableView(tableView: UITableView, cellForRowAtIndexPath
indexPath: NSIndexPath) -> UITableViewCell
    {
        let cell = tableView.dequeueReusableCellWithIdentifier("myCell",
forIndexPath: indexPath) as! UITableViewCell

        // Configure the cell...
        cell.textLabel!.text = "row: \(indexPath.row)"

        if cell.selected
        {
            cell.accessoryType = UITableViewCellAccessoryType.Checkmark
        }
        else
        {
            cell.accessoryType = UITableViewCellAccessoryType.None
        }

        return cell
    }

    override func tableView(tableView: UITableView, didSelectRowAtIndexPath
indexPath: NSIndexPath)
    {
        let cell = tableView.cellForRowAtIndexPath(indexPath)

        if cell!.selected == true
        {
            cell!.accessoryType = UITableViewCellAccessoryType.Checkmark
        }
        else
        {
            cell!.accessoryType = UITableViewCellAccessoryType.None
        }
    }

    override func tableView(tableView: UITableView, numberOfRowsInSection
section: Int) -> Int
    {
        return 100
    }
}
```

I made a sample project here: https://github.com/brcimo/SwiftTableViewMultipleSelection

Share Follow

answered Jun 2, 2015 at 22:56

Bryan Cimo

1,268 1 19 34

2 This doesn't seem to work. `tableView:didSelectRowAtIndexPath:` get called only when the row is selected, not when is deselecting, hence `cell.selected` will always be true! – entropid Nov 11,

2015 at 0:41