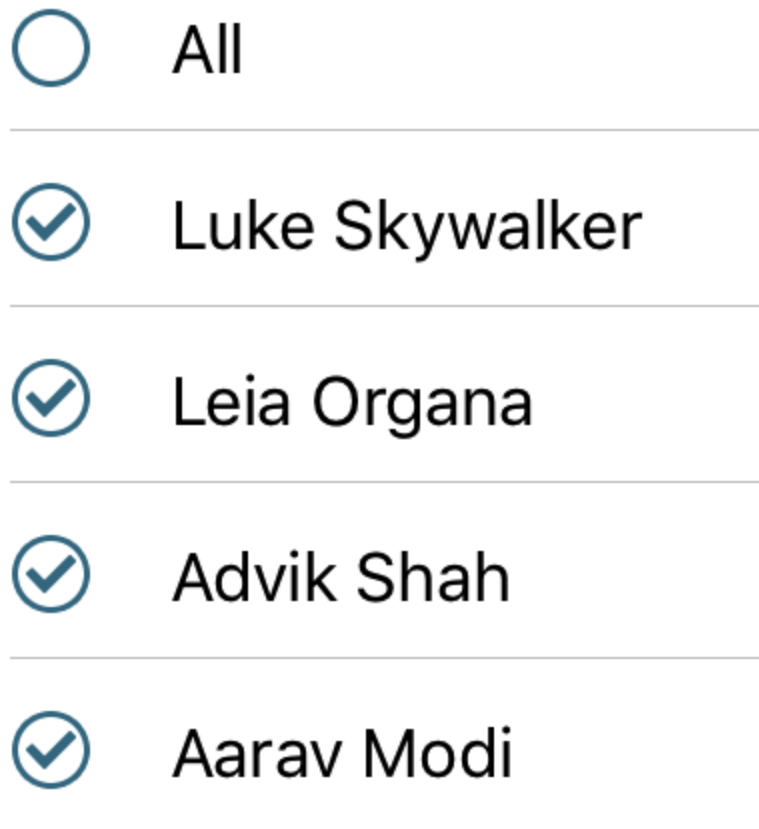


# Swift 3.0 multiple selection with select all cell

Asked 6 years, 1 month ago Modified 3 years, 9 months ago Viewed 16k times

Part of [Mobile Development](#) Collective

I have added data in table view and I have manually added "select all" option to the list at first position, now when the user selects the first option which is 'select all' then the person manually option "Select all" is not selected. Select all, click then work all person or deselect working but signal selection all the person not working "Select all" I have tried the code below but it's not working so can any one help me to solve this?



```
var unchecked:Bool = true
func tableView(_ tableView: UITableView, cellForRowAt indexPath:
IndexPath) -> UITableViewCell {
    // create a new cell if needed or reuse an old one
    let cell = ObjTableview.dequeueReusableCell(withIdentifier: "Cell",
for: indexPath) as! SelectUserCell
    // set the text from the data model
    cell.selectionStyle = UITableViewCellStyle.none
    cell.lblStudentName.text = getStudentName[indexPath.row]

    if UnAll == "unselect" {
        if indexPath.row == 0 {
            cell.btnCheckbox.setImage(UIImage(named: "unSelectedItem"),
for: .normal)
```

```

    }
    if indexPath.row == Int(selectedNumber) {
        cell.btnCheckbox.setImage(UIImage(named: "unSelectedItem"),
for: .normal)

    }
    if indexPath.row == Int(unSelectNumber) {
        //var j = "\(i)"

        cell.btnCheckbox.setImage(UIImage(named: "selectedItem"),
for: .normal)

    }

}
}else
{
    if(unchecked){

        cell.btnCheckbox.setImage(UIImage(named: "unSelectedItem"),
for: .normal)

    }
    else{

        cell.btnCheckbox.setImage(UIImage(named: "selectedItem"), for:
.normal)

    }
}

}

return cell
}
var UnAll = ""
var selectedNumber = ""
var unSelectNumber = ""
var checkselect:Bool = true

func tableView(_ tableView: UITableView, didSelectRowAt indexPath:
IndexPath) {

    UnAll.removeAll()
    selectedNumber.removeAll()
    unSelectNumber.removeAll()
    if(indexPath.row == 0){

        btnCheckBoxClick(sender: UIButton())

    }else
    {
        UnAll = "unselect"
        btnCheckBoxClick(sender: UIButton())
        if checkselect {
            selectedNumber = "\(indexPath.row)"
            checkselect = false
        }else
        {
            unSelectNumber = "\(indexPath.row)"
            checkselect = true
        }

        print("the selected index is : \(indexPath.row)")
    }
}

```

```

}

@IBAction func btnCheckBoxClick(_ sender: Any) {

    if(unchecked){

        unchecked = false
    }
    else{
        unchecked = true
    }
    ObjTableView.reloadData()
}

```

ios   iphone   uitableview   swift3

Share Follow

edited May 18, 2018 at 8:54



Elydasian

2,026 5 23 41

asked Sep 27, 2017 at 5:21



Jayprakash Singh

1,363 3 16 28

@RajeshkumarR yes – Jayprakash Singh Sep 27, 2017 at 5:43

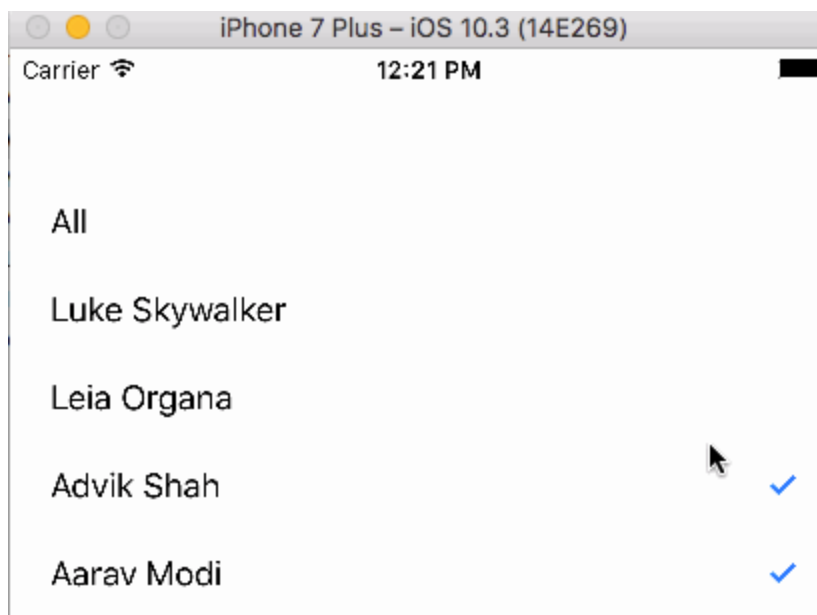
## 4 Answers

Sorted by: Highest score (default)



Create a struct for model data with a Bool property. You can modify this property by cell selection.

9



```

class ViewController: UIViewController, UITableViewDelegate,
UITableViewDataSource {

```

```

var allCharacters:[Character] = []

override func viewDidLoad() {
    super.viewDidLoad()
    allCharacters = [Character(name: "All"),Character(name: "Luke
Skywalker"),Character(name: "Leia Organa"),Character(name: "Advik
Shah"),Character(name: "Aarav Modi")]
}
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int)
-> Int {
    return allCharacters.count
}
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)
-> UITableViewCell {
    var cell = tableView.dequeueReusableCell(withIdentifier: "Cell")
    if cell == nil{
        cell = UITableViewCell(style: .subtitle, reuseIdentifier: "Cell")
    }
    cell?.textLabel?.text = allCharacters[indexPath.row].name
    if allCharacters[indexPath.row].isSelected
    {
        cell?.accessoryType = .checkmark
    }
    else
    {
        cell?.accessoryType = .none
    }
    cell?.selectionStyle = .none
    return cell!
}
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath)
{
    if indexPath.row == 0
    {
        allCharacters[indexPath.row].isSelected =
!allCharacters[indexPath.row].isSelected
        for index in allCharacters.indices
        {
            allCharacters[index].isSelected =
allCharacters[indexPath.row].isSelected
        }
    }
    else
    {
        allCharacters[indexPath.row].isSelected =
!allCharacters[indexPath.row].isSelected
        if allCharacters.dropFirst().filter({ $0.isSelected }).count ==
allCharacters.dropFirst().count
        {
            allCharacters[0].isSelected = true
        }
        else
        {
            allCharacters[0].isSelected = false
        }
    }
    tableView.reloadData()
}

```

```

}

struct Character
{
    var name:String
    // var otherDetails
    var isSelected:Bool! = false
    init(name:String) {
        self.name = name
    }
}

```

## Creating Array of Struct objects from array of dictionary

```

let SubjectArray = json["students"] as! [[String:Any]]
allCharacters = SubjectArray.map({ Character(name: $0["studentName"] as!
String) })
allCharacters.insert(Character(name:"All"), at: 0)

```

Share Follow

edited Dec 29, 2017 at 5:33

answered Sep 27, 2017 at 6:22



**RajeshKumar R**

15.5k 2 38 70

@Rajeshkumar my data come with api through how can be add tableview like allCharacters  
– Jayprakash Singh Sep 27, 2017 at 6:39

@Rajeshkumar R how to add struct in api data – Jayprakash Singh Sep 27, 2017 at 7:27

@Rajeshkumar R not working error :- Cannot convert call result type '\_' to expected type  
'[Character]' – Jayprakash Singh Sep 27, 2017 at 7:39

Thanks for discussion – Jayprakash Singh Sep 27, 2017 at 10:20

It saves my time.Thank you @RajeshKumar R – Preetha May 18, 2018 at 9:26



4



I like @Pranil's suggestion of using a separate section for the "All" row, so I have stolen that.

You can use an `NSMutableIndexSet` for tracking the selected rows. This is simpler than having to create a new struct or array of booleans or something. The only thing you do need to be aware of is if your tableview allows row reordering then the index set needs to be adjusted accordingly.

Here is my implementation. The "all" state is determined by the number of selected rows being equal to the number of rows in the data source array.

I have just used simple table view accessories for the checkmarks, but I am sure you can see how to adopt your image based approach in `cellForRow(at:)`

```
import UIKit
```

```
class ViewController: UIViewController, UITableViewDataSource,
UITableViewDelegate {

    @IBOutlet weak var tableView: UITableView!

    let names: [String]? = ["Luke Skywalker", "Leia Organa", "Advik Shah", "Aarav
Modi"]

    var selectedRows = NSMutableIndexSet()

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a
nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    func numberOfSections(in tableView: UITableView) -> Int {

        return 2
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section:
Int) -> Int {

        guard let names = self.names else {
            return 0
        }

        return 0 == section ? 1 : names.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)
-> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "tableCell",
for: indexPath)

        var text: String
        var accessory = UITableViewCellAccessoryType.none

        if 0 == indexPath.section {
            text = "All"
            if self.selectedRows.count == self.names!.count {
                accessory = .checkmark
            }
        } else {
            text = names![indexPath.row]
            if selectedRows.contains(indexPath.row) {
                accessory = .checkmark
            }
        }

        cell.textLabel!.text = text
        cell.accessoryType = accessory
    }
}
```

```

        return cell
    }

    func tableView(_ tableView: UITableView, willSelectRowAt indexPath:
IndexPath) -> IndexPath? {
        if indexPath.section == 0 {
            if self.selectedRows.count == self.names!.count {
                self.selectedRows = NSMutableIndexSet()
            } else {
                self.selectedRows = NSMutableIndexSet(indexesIn:
NSRange(location: 0, length: self.names!.count))
            }
            tableView.reloadData()
        } else {
            self.selectedRows.contains(indexPath.row) ?
self.selectedRows.remove(indexPath.row) : self.selectedRows.add(indexPath.row)

            let rows = [IndexPath(row: 0, section: 0), indexPath]

            tableView.reloadRows(at: rows, with: .none)
        }
        return nil
    }
}

```

Share Follow

answered Sep 27, 2017 at 8:02



Paulw11

110k 14 163 191



3



I think you are using only one section in the table view. I suggest you use two sections in the table view, so that first section will contain only one row (Select All) and the second section will contain other options. When you click on Select All, that is in the first row of the first section you can make all the rows in the second section as selected while reloading the table view.



```

// MARK: - struct for cell item
struct CellItem {
    var name : String
    var isSelected:Bool! = false
    init(name: String) {
        self.name = name
    }
}

```

```

class ViewController: UITableViewController {

    @IBOutlet var viewTable: UITableView!
    // Declare a boolean variable to toggle the checkbox in the first section of
    table view
    var isSelectAllSelected : Bool = false
    var cellData: [CellItem] = []

    override func viewDidLoad() {
        super.viewDidLoad()
    }
}

```

```

        cellData = [CellItem(name: "Luke Skywalker"), CellItem(name: "Leia
Organa"), CellItem(name: "Advik Shah"), CellItem(name: "Aarav Modi")]
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }

    // MARK: - Table view data source

    override func numberOfSections(in tableView: UITableView) -> Int {
        return 2
    }

    override func tableView(_ tableView: UITableView, numberOfRowsInSectionSection: Int) -> Int {
        if section == 0 {
            return 1
        }
        else
        {
            return cellData.count
        }
    }

    override func tableView(_ tableView: UITableView, heightForHeaderInSection: Int) -> CGFloat {
        return 0
    }

    // MARK: - Table view delegates

    override func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
        return 60
    }

    override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = TableCell()
        cell.selectionStyle = .none

        if indexPath.section == 0 {
            cell.textLabel?.text = "Select All"
            if isSelectedAllSelected {
                cell.accessoryType = .checkmark
            }
            else {
                cell.accessoryType = .none
            }
        }
        else
        {
            cell.textLabel?.text = cellData[indexPath.row].name
            if cellData[indexPath.row].isSelected {
                cell.accessoryType = .checkmark
            }
            else {
                cell.accessoryType = .none
            }
        }
    }

```



```

    }
    return cell
}

override func tableView(_ tableView: UITableView, didSelectRowAt indexPath:
IndexPath) {
    if indexPath.section == 0
    {
        cellData[indexPath.row].isSelected =
!cellData[indexPath.row].isSelected
        isSelectAllSelected = cellData[indexPath.row].isSelected
        for index in cellData.indices
        {
            cellData[index].isSelected = cellData[indexPath.row].isSelected
        }
    }
    else
    {
        cellData[indexPath.row].isSelected =
!cellData[indexPath.row].isSelected
        if cellData.filter({ $0.isSelected }).count == cellData.count
        {
            isSelectAllSelected = true
        }
        else
        {
            isSelectAllSelected = false
        }
    }
    viewTable.reloadData()
} }

```

Share Follow

edited Sep 28, 2017 at 7:53

answered Sep 27, 2017 at 6:17



iPran

126 5

@ pranil any example ..? – Jayprakash Singh Sep 27, 2017 at 6:20

@JayprakashSingh I have edited my answer to include table view data sources and delegates to implement the logic using table view sections. – iPran Sep 28, 2017 at 5:54

Guys, please refer the GitHub link in which I have shown how exactly we can use table view sections to implement multiple selections with a select all cell: [github.com/iPran/TableView-MultipleSelection](https://github.com/iPran/TableView-MultipleSelection) – iPran Nov 13, 2017 at 13:01



2

Hello u can take cheboxbutton action method inside view controller with addtarget method and assign tag indxpath.row so u can easily get the indxpath. from below code u can get the idea.



```

class ViewController:UIViewController,UITableViewDelegate,UITableViewDataSource
{

```



```

@IBOutlet weak var ObjTableview: UITableView!
var arrStudent = ["1","2","3","4","5"]
var arrSelectedStudent :[Int] = []
var selectAll:Bool = false

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
    // Dispose of any resources that can be recreated.
}

//MARK: UITableViewDataSource
func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) ->
Int {
    return arrStudent.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {
    // create a new cell if needed or reuse an old one
    let cell = ObjTableview.dequeueReusableCell(withIdentifier: "Cell", for:
indexPath) as! SelectUserCell
    // set the text from the data model
    cell.selectionStyle = UITableViewCellStyle.none
    // cell.lblStudentName.text = getStudentName[indexPath.row]
    cell.lblStudentName.text = arrStudent[indexPath.row]
    cell.btnCheckbox.tag = indexPath.row
    cell.btnCheckbox.addTarget(self,
action:#selector(btnCheckBoxClick(sender:)), for: .touchUpInside)

    if selectAll {
        cell.btnCheckbox.setImage(UIImage(named: "selectedItem"), for: .normal)
    }else{
        if arrSelectedStudent.contains(indexPath.row){
            cell.btnCheckbox.setImage(UIImage(named: "selectedItem"), for: .normal)
        }else{
            cell.btnCheckbox.setImage(UIImage(named: "unSelectedItem"), for:
.normal)
        }
    }

    return cell
}

func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
}

func btnCheckBoxClick(sender: UIButton) {
    if sender.tag == 0{
        selectAll = true
    }else
    {
        selectAll = false
    }
    if let index = arrSelectedStudent.index(of: sender.tag) {

```

```
        arrSelectedStudent.remove(at: index)
    }else{
        arrSelectedStudent.append(sender.tag)
    }
}

ObjTableView.reloadData()
}}
```

Share Follow

answered Sep 27, 2017 at 7:47



[pradip rathod](#)

**348** 3 20