

# **UISearchController**

## **Tutorial: Getting Started**

//UISearchController 튜토리얼 Ray Wenderlich 사이트

[//https://www.raywenderlich.com/472-uisearchcontroller-tutorial-getting-started](https://www.raywenderlich.com/472-uisearchcontroller-tutorial-getting-started)

이 UISearchController 자습서에서는 표준 테이블 뷰를 기반으로 검색 가능한 캔디 앱을 작성합니다. 테이블 뷰 검색 기능과 동적 필터링 기능을 추가하고 선택 사항인 스코프 바를 추가하여 모든 기능을 활용할 수 UISearchController 있습니다. 결국 앱을 훨씬 더 사용자 친

# Getting Started














Download the starter project from the tutorial [here](#) and open the project.



3:04



## Candy Search

- ▼  CandySearch
  - ▼  CandySearch
    -  AppDelegate.swift
    -  CandyTheme.swift
    -  MasterViewController.swift
    -  SearchFooter.swift
    -  DetailViewController.swift
    -  Candy.swift
    -  Main.storyboard
    -  Assets.xcassets
    -  LaunchScreen.storyboard
    -  Info.plist
  - ▶  Products

```
import Foundation
```

```
//Candy의 각 조각에 대한 정보를 저장하는 구조체
```


```
struct Candy {  
    let category : String  
    let name : String  
}
```



## 테이블 뷰 채우기

**MasterViewController.swift**를 엽니다. 이 `candies` 속성은 `Candy` 사용자가 검색 할 수 있는 모든 개체를 관리하는 곳입니다. 말하자면, 이제는 사탕을 만들 때입니다!





▼  CandySearch


▼  CandySearch


 AppDelegate.swift


 CandyTheme.swift


 MasterViewController.swift


 SearchFooter.swift


 DetailViewController.swift


 Candy.swift

 Main.storyboard

 Assets.xcassets

 LaunchScreen.storyboard

 Info.plist

▶  Products

```
// MARK: - View Setup
override func viewDidLoad() {
    super.viewDidLoad()
    candies = [
        Candy(category:"Chocolate", name:"Chocolate Bar"),
        Candy(category:"Chocolate", name:"Chocolate Chip"),
        Candy(category:"Chocolate", name:"Dark Chocolate"),
        Candy(category:"Hard", name:"Lollipop"),
        Candy(category:"Hard", name:"Candy Cane"),
        Candy(category:"Hard", name:"Jaw Breaker"),
        Candy(category:"Other", name:"Caramel"),
        Candy(category:"Other", name:"Sour Chew"),
        Candy(category:"Other", name:"Gummi Bear"),
        Candy(category:"Other", name:"Candy Floss"),
        Candy(category:"Chocolate", name:"Chocolate Coin"),
        Candy(category:"Chocolate", name:"Chocolate Egg"),
        Candy(category:"Other", name:"Jelly Beans"),
        Candy(category:"Other", name:"Liquorice"),
        Candy(category:"Hard", name:"Toffee Apple")
    ]
}
```



3:07



## Candy Search

Chocolate Bar

Chocolate

Chocolate Chip

Chocolate

Dark Chocolate

Chocolate

Lollipop

Hard

Candy Cane

Hard

Jaw Breaker

Hard

Caramel

Other

Sour Chew

Other

Gummi Bear

Other

Candy Floss

Other

Chocolate Coin

Chocolate

Chocolate Egg

Chocolate

Jelly Beans

Other

Liquorice

Other

Toffee Apple

Hard

---

---

---

---

# Chocolate Bar

## Chocolate

---

© 2014 The Hershey Company. All rights reserved. [www.hersheys.com](#)

3:07



[← Back](#)

# Chocolate

Chocolate Bar



# UISearchController 소개

`UISearchController` 문서 를 보면 꽤 게으르다는 것을 알 수 있습니다. 그것은 전혀 검색 작업을 수행하지 않습니다. 이 클래스는 사용자가 iOS 앱에서 기대하는 표준 인터페이스를 제공하기 만하면됩니다.



```
class MasterViewController: UIViewController, UITableViewDataSource, UITableViewDelegate {  
  
    // MARK: - Properties  
    @IBOutlet var tableView: UITableView!  
    @IBOutlet var searchFooter: SearchFooter!  
  
    var detailViewController: DetailViewController? = nil  
    var candies = [Candy]()  
    let searchController = UISearchController(searchResultsController: nil)
```

```
}  
}
```

```
//클래스 바깥에 extension 추가
```

```
extension MasterViewController: UISearchResultsUpdating {  
    // MARK: - UISearchResultsUpdating Delegate  
    func updateSearchResults(for searchController: UISearchController) {  
        // TODO  
    }  
}
```

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

```
// 검색 컨트롤러를 설정합니다.
```

```
searchController.searchResultsUpdater = self  
searchController.obscuresBackgroundDuringPresentation = false  
searchController.searchBar.placeholder = "Search Candies"  
navigationItem.searchController = searchController  
definesPresentationContext = true
```

```
candies = [  
    Candy(category:"Chocolate", name:"Chocolate Bar"),
```

```
var detailViewController: DetailViewController? = nil
```

```
//사용자가 검색중인 사탕을 보유
```

```
var filteredCandies = [Candy]()
```

```
var candies = [Candy]()
```

```
let searchController = UISearchController(searchResultsController: nil)
```

```
// MARK: – Private instance methods
```

```
func searchBarIsEmpty() -> Bool {  
    // Returns true if the text is empty or nil  
    return searchController.searchBar.text?.isEmpty ?? true  
}
```

```
func filterContentForSearchText(_ searchText: String, scope: String = "All") {  
    filteredCandies = candies.filter({( candy : Candy) -> Bool in  
        return candy.name.lowercased().contains(searchText.lowercased())  
    })  
  
    tableView.reloadData()  
}
```

```
// MARK: – View Setup
```

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

//클래스 바깥에 extension 추가

```
extension MasterViewController: UISearchResultsUpdating {  
    // MARK: - UISearchResultsUpdating Delegate  
    func updateSearchResults(for searchController: UISearchController) {  
        //TO DO  
        filterContentForSearchText(searchController.searchBar.text!)  
    }  
}
```



3:27



## Candy Search

🔍 Search Candies

Chocolate Bar

Chocolate

Chocolate Chip

Chocolate

Dark Chocolate

Chocolate

Lollipop

Hard

Candy Cane

Hard

Jaw Breaker

Hard

Caramel

Other

Sour Chew

Other

Gummi Bear

Other

Candy Floss

Other

Chocolate Coin

Chocolate

Chocolate Egg

Chocolate

Jelly Beans

Other

Liquorice

Other

Toffee Apple

Hard



# Updating the Table View

```
func isFiltering() -> Bool {  
    return searchController.isActive && !searchBarIsEmpty()  
}
```

```
// MARK: - View Setup  
override func viewDidLoad() {  
    super.viewDidLoad()
```

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    if isFiltering() {  
        return filteredCandies.count  
    }  
    return candies.count  
}
```

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

    //let candy = candies[indexPath.row]
    let candy: Candy
    if isFiltering() {
        candy = filteredCandies[indexPath.row]
    } else {
        candy = candies[indexPath.row]
    }

    cell.textLabel!.text = candy.name
    cell.detailTextLabel!.text = candy.category
    return cell
}
```



3:31



🔍 Cho



Cancel

Chocolate Bar

Chocolate

Chocolate Chip

Chocolate

Dark Chocolate

Chocolate

Chocolate Coin

Chocolate

Chocolate Egg

Chocolate

# **Sending Data to a Detail View**

```
// MARK: - Segues
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "showDetail" {
        if let indexPath = tableView.indexPathForSelectedRow {
            //let candy = candies[indexPath.row]
            let candy: Candy
            if isFiltering() {
                candy = filteredCandies[indexPath.row]
            } else {
                candy = candies[indexPath.row]
            }

            let controller = (segue.destination as! UINavigationController).topViewController as!
                DetailViewController
            controller.detailCandy = candy
            controller.navigationItem.leftBarButtonItem = splitViewController?.displayModeButtonItem
            controller.navigationItem.leftItemsSupplementBackButton = true
        }
    }
}
```





// MARK: - Segues

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "showDetail" {
        if let indexPath = tableView.indexPathForSelectedRow {
            //검색한 후에 행을 선택해서 Detail View로 전환할 때 filtered Candies에서 선택한 행르 제대로 보여주도록 수정한다.
            //수정하지 않으면 원본 Candy 배열에서 행을 선택해서 보여주므로 잘못된 것이 보인다.
            //let candy = candies[indexPath.row]
            let candy: Candy
            if isFiltering() {
                candy = filteredCandies[indexPath.row]
            } else {
                candy = candies[indexPath.row]
            }

            let controller = (segue.destination as! UINavigationController).topViewController as!
                DetailViewController
            controller.detailCandy = candy
            controller.navigationItem.leftBarButtonItem = splitViewController?.displayModeButtonItem
            controller.navigationItem.leftItemsSupplementBackButton = true
        }
    }
}
```



Cho



Cancel

Chocolate Bar

Chocolate

Chocolate Chip

Chocolate

Dark Chocolate

Chocolate

Chocolate Coin

Chocolate

Chocolate Egg

Chocolate

---

# Chocolate Egg

## Chocolate

---

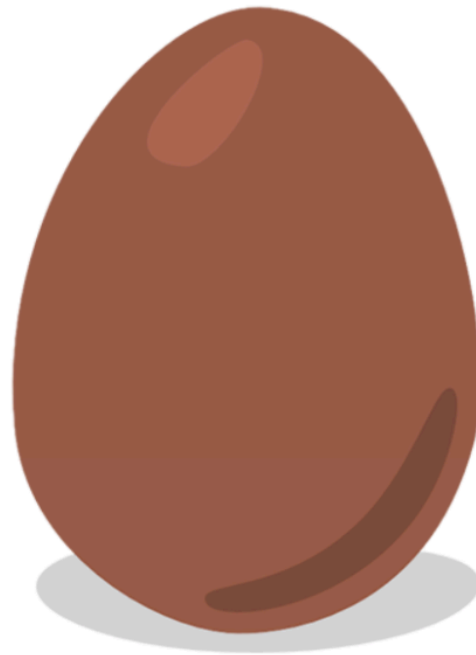
3:38



[← Back](#)

# Chocolate

Chocolate Egg



# **Creating a Scope Bar to Filter Results**

//클래스 바깥에 extension 추가

```
extension MasterViewController: UISearchResultsUpdating {  
    // MARK: - UISearchResultsUpdating Delegate  
    func updateSearchResults(for searchController: UISearchController) {  
        //TO DO  
        filterContentForSearchText(searchController.searchBar.text!)  
    }  
}
```

//새로운 extension

```
extension MasterViewController: UISearchBarDelegate {  
    // MARK: - UISearchBar Delegate  
    func searchBar(_ searchBar: UISearchBar, selectedScopeButtonIndexDidChange selectedScope: Int) {  
        filterContentForSearchText(searchBar.text!, scope: searchBar.scopeButtonTitles!  
            [selectedScope])  
    }  
}
```

```
func filterContentForSearchText(_ searchText: String, scope: String = "All") {
    filteredCandies = candies.filter({( candy : Candy) -> Bool in
        // return candy.name.lowercased().contains(searchText.lowercased())
        let doesCategoryMatch = (scope == "All") || (candy.category == scope)

        if searchBarIsEmpty() {
            return doesCategoryMatch
        } else {
            return doesCategoryMatch && candy.name.lowercased().contains(searchText.lowercased())
        }
    })
    tableView.reloadData()
}

func isFiltering() -> Bool {
    return searchController.isActive && !searchBarIsEmpty()
}

// MARK: - View Setup
override func viewDidLoad() {
```



```
-  
func isFiltering() -> Bool {  
    //return searchController.isActive && !searchBarIsEmpty()  
    let searchBarScopeIsFiltering = searchController.searchBar.selectedScopeButtonIndex != 0  
    return searchController.isActive && (!searchBarIsEmpty() || searchBarScopeIsFiltering)  
}  
// MARK: - View Setup  
override func viewDidLoad() {  
    super.viewDidLoad()
```

//클래스 바깥에 extension 추가

```
extension MasterViewController: UISearchResultsUpdating {  
    // MARK: - UISearchResultsUpdating Delegate  
    func updateSearchResults(for searchController: UISearchController) {  
        //TO DO  
        //filterContentForSearchText(searchController.searchBar.text!)  
        let searchBar = searchController.searchBar  
        let scope = searchBar.scopeButtonTitles![searchBar.selectedScopeButtonIndex]  
        filterContentForSearchText(searchController.searchBar.text!, scope: scope)  
    }  
}
```

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

```
    // 검색 컨트롤러를 설정합니다.
```

```
    searchController.searchResultsUpdater = self  
    searchController.obscuresBackgroundDuringPresentation = false  
    searchController.searchBar.placeholder = "Search Candies"  
    navigationItem.searchController = searchController  
    definesPresentationContext = true
```

```
    // Setup the Scope Bar
```

```
    searchController.searchBar.scopeButtonTitles = ["All", "Chocolate", "Hard", "Other"]  
    searchController.searchBar.delegate = self  
    candies = [  
        Candy(category:"Chocolate", name:"Chocolate Bar"),
```



3:44



🔍 Search Candies

Cancel

All

Chocolate

Hard

Other

Lollipop

Hard

Candy Cane

Hard

Jaw Breaker

Hard

Toffee Apple

Hard

# **Adding a Results Indicator**

```
// Setup the Scope Bar
searchController.searchBar.scopeButtonTitles = ["All", "Chocolate", "Hard", "Other"]
searchController.searchBar.delegate = self
// Setup the search footer
tableView.tableFooterView = searchFooter
candies = [
    Candy(category:"Chocolate", name:"Chocolate Bar"),
```

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    if isFiltering() {  
        //return filteredCandies.count  
        searchFooter.setIsFilteringToShow(filteredItemCount: filteredCandies.count, of: candies.count)  
        return filteredCandies.count  
    }  
    searchFooter.setNotFiltering()  
    return candies.count  
}
```





3:48



Ch



Cancel

All

Chocolate

Hard

Other

Chocolate Bar

Chocolate

Chocolate Chip

Chocolate

Dark Chocolate

Chocolate

Sour Chew

Other

Chocolate Coin

Chocolate

Chocolate Egg

Chocolate

Filtering 6 of 15