Documentation

Language: **Swift**

UIKit / View controllers / Using suggested searches with a search controller

Sample Code

Using suggested searches with a search controller

Create a search interface with a table view of suggested searches.

Download

iOS 13.2+ | iPadOS 13.2+ | Xcode 13.0+

Overview

This sample project demonstrates how to implement a `UISearchController` in your application. `UISearchController` is a view controller that manages the display of search results based on interactions with a search bar.

The sample lists flower products. Users can search for them by title, price, year, and color. As soon as the user taps the search bar, the search controller displays the list of suggested searches. A suggested search represents a particular color: red, blue, or green. The search controller filters the list of flower products by color when a suggested color is selected. This behavior matches that of the Mail app.

This sample also demonstrates a scene-based architecture using `UISceneDelegate`. It includes the recommended `UIUserActivityRestoring` protocol to save and restore the search results. Adopt this protocol to save the search bar's active state, first responder status, search bar text and token, and restore them when the app is relaunched.

Create a search controller

The `MainTableViewController`, a subclass of `UITableViewController`, creates the search controller. The search controller's search bar helps filter a set of `Product` objects and displays the results in the table view. The sample places the search controller in the `MainTableViewController's` navigation bar:

```
searchController = UISearchController(searchResultsController: resultsTableControlle
searchController.searchResultsUpdater = self
searchController.searchBar.autocapitalizationType = .none
searchController.searchBar.searchTextField.placeholder = NSLocalizedString("Enter a
searchController.searchBar.returnKeyType = .done

// Place the search bar in the navigation bar.
navigationItem.searchController = searchController

// Make the search bar always visible.
navigationItem.hidesSearchBarWhenScrolling = false

// Monitor when the search controller is presented and dismissed.
searchController.delegate = self

// Monitor when the search button is tapped, and start/end editing.
searchController.searchBar.delegate = self
```

The `ResultsTableController` displays both the list of filtered flower products and the list of suggested search colors. As the user types text in the search field, the list of flower products matches what is typed. If the user doesn't enter any text, the search controller displays a list of suggested search colors. A suggested search represents a single query to search for a specific color of a product.

Create suggested searches

The list of suggested searches is displayed by the results controller once the search bar is tapped:

```
    func presentSearchController(_ searchController: UISearchController) {
        searchController.showsSearchResultsController = true
        setToSuggestedSearches()
}
```

When the user first taps the search bar, the search results controller displays the suggested searches. The user chooses a suggested search, and then types additional search criteria in the search bar. Each suggested search represents a `UISearchToken`, a visual representation of a search query. Tapping a suggested search creates a search token for a particular color and the search controller places it in the search bar's text field. This text field is represented by `UISearchTextField` which supports cut, copy, paste, and drag and drop of search tokens. Tokens always precede the text and can be selected and deleted by the user.

The sample creates a `UISearchToken` as follows:

```
class func searchToken(tokenValue: Int) -> UISearchToken {
    let tokenColor = ResultsTableController.suggestedColor(fromIndex: tokenValue)
    let image =
        UIImage(systemName: "circle.fill")?.withTintColor(tokenColor, renderingMode:
    let searchToken = UISearchToken(icon: image, text: suggestedTitle(fromIndex: tok

    // Set the color kind number as the token value.
    let color = ResultsTableController.colorKind(fromIndex: tokenValue).rawValue
    searchToken.representedObject = NSNumber(value: color)

    return searchToken
}
```

This sample inserts the search token into the search bar's search text field as follows:

```
if let searchField = navigationItem.searchController?.searchBar.searchTextField {
    searchField.insertToken(token, at: 0)

    // Hide the suggested searches now that we have a token.
    resultsTableController.showSuggestedSearches = false

    // Update the search query with the newly inserted token.
    updateSearchResults(for: searchController)
}
```

See Also

Search interface

`class UISearchContainerViewController`

A view controller that manages the presentation of search results in your interface.

`class UISearchController`

A view controller that manages the display of search results based on interactions with a search bar.

`class UISearchBar`

A specialized view for receiving search-related information from the user.

`protocol UISearchResultsUpdating`

A set of methods that let you update search results based on information the user enters into the search bar.

{} Displaying searchable content by using a search controller

Create a user interface with searchable content in a table view.