Toolbar Actions

Sharing a Recipe from the Toolbar

The previous tutorial added standard and custom items to the toolbar. In this tutorial, you'll add a service picker item that enables users to share recipes with others.

Download the project files, and follow the steps to share a recipe from the toolbar.

30mins
Estimated Time

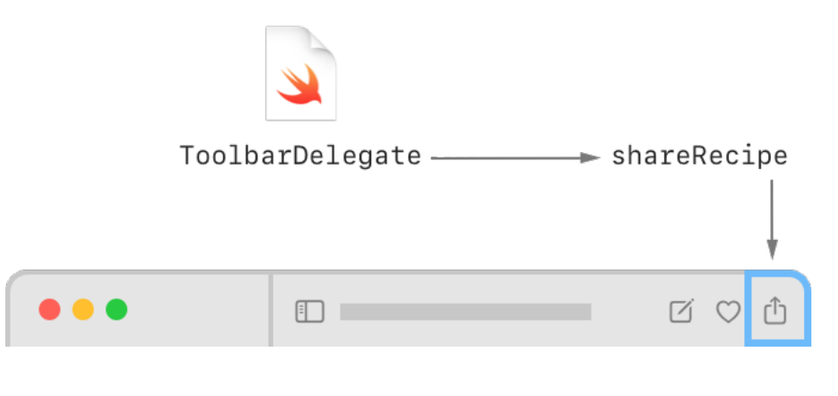
Project files

Xcode 12.2 or later

Section 1

Add a Share Recipe Item to the Toolbar

The last item you'll add to the toolbar provides users the ability to share a recipe. You'll create an instance of `NSSharingServicePickerToolbarItem` to display a list of Share options when the user clicks the toolbar item. You'll also update the app to create a `UIActivityItemsConfiguration` object that tells the Share Recipe toolbar item which recipe to share.



The toolbar delegate needs to keep a reference to the Share Recipe item added to the toolbar so the app can update the item's activity item configuration every time the user selects a recipe. The activity item configuration tells the item which recipe to share.

Step 1

Select `ToolbarDelegate.swift` to open the file in the editor and add the property `shareRecipe` of type `NSSharingServicePickerToolbarItem` to the delegate. You'll use this property in Step 6 to store a temporary reference to the Share Recipe toolbar item.

Note

The class `NSSharingServicePickerToolbarItem` is only available to apps built with Mac Catalyst, so enclose the property declaration in a `targetEnvironment(macCatalyst)` conditional block.

ToolbarDelegate.swift

No Preview

```swift
import UIKit

class ToolbarDelegate: NSObject {

    #if targetEnvironment(macCatalyst)
    var shareRecipe: NSSharingServicePickerToolbarItem?
    #endif

}

#if targetEnvironment(macCatalyst)
extension NSToolbarItem.Identifier {
    static let editRecipe = NSToolbarItem.Identifier("com.example.apple-samplecode.Recipes.e
    static let toggleRecipeIsFavorite = NSToolbarItem.Identifier("com.example.apple-sampleco
}

extension ToolbarDelegate: NSToolbarDelegate {

    func toolbarDefaultItemIdentifiers(_ toolbar: NSToolbar) -> [NSToolbarItem.Identifier] {
        let identifiers: [NSToolbarItem.Identifier] = [
            .toggleSidebar,
            .flexibleSpace,
            .editRecipe,
            .toggleRecipeIsFavorite
        ]
        return identifiers
    }

```

Next, add the custom identifier for the Share Recipe item and return a toolbar item instance when the toolbar asks for it.

Step 2

Add the `shareRecipe` toolbar item identifier to the extension containing the other custom identifiers

Step 3

Add the `shareRecipe` item identifier to the list of default toolbar item identifiers.

Step 4

Now add a case statement to the `toolbar(_:itemForItemIdentifier:willBeInsertedIntoToolbar:)` method that returns a new instance of `NSSharingServicePickerToolbarItem`.

Save and remove the temporary reference to the Share Recipe toolbar item as the toolbar adds and removes the item.

Step 5

Scroll the editor to the bottom of `ToolbarDelegate.swift`, and add the delegate method `toolbarWillAddItem(_:)`.

The system calls `toolbarWillAddItem(_:)` before adding a new item to a toolbar, making this method the ideal place to store a reference to the toolbar item. Store a toolbar item reference only if the app needs to make changes to the item after it appears in the toolbar; for example, store the reference if the app needs to set the activity item configuration of the toolbar item after the user selects the recipe.

Step 6

In the implementation of the `toolbarWillAddItem(_:)` method, retrieve the Share Recipe toolbar item from the incoming notification's `userInfo` dictionary and store the reference to the item in the property `shareRecipe`.

Step 7

Lastly, add the method `toolbarDidRemoveItem(_:)` and set `shareRecipe` to `nil` when the toolbar removes the Share Recipe item.

Important

The Recipes app doesn't display a user customizable toolbar, but if it did, it would be possible for the user to remove the Share Recipe item from the toolbar. For this reason, whenever you implement `toolbarWillAddItem(_:)` to store a reference to a toolbar item, you should also implement `toolbarDidRemoveItem(_:)` to remove that reference.

Section 2

Indicate a Recipe to Share

The Share Recipe item now appears in the toolbar, but it doesn't know which recipe to share. The toolbar item needs a reference to an activity items configuration that contains the recipe, and the app needs to update the reference to the activity items configuration each time the user selects another recipe.

`RecipeDetailViewController` includes a reference to the selected recipe, so it's the ideal class to create the activity item configuration each time the user selects a recipe.

Step 1

In the Project navigator, select `RecipeDetailViewController.swift` to open the file in the editor. Then scroll to the end of the file and add a new class extension.

RecipeDetailViewController.swift

No Preview

```swift
import UIKit
import os
import Combine

class RecipeDetailViewController: UIViewController, RecipeController {

    static let storyboardID = "RecipeDetail"
    static func instantiateFromStoryboard() -> RecipeDetailViewController? {
        let storyboard = UIStoryboard(name: "RecipeDetail", bundle: .main)
        return storyboard.instantiateViewController(identifier: storyboardID) as? RecipeDetail
    }

    var recipe: Recipe? {
        didSet {
            updateUI()
        }
    }

    @IBOutlet weak var recipeFavoriteButton: UIBarButtonItem!

    private var topChildController: RecipeController?
    private var bottomChildController: RecipeController?

    var noRecipeView: UIView!

    private var dataStoreSubscriber: AnyCancellable?

    override func viewDidLoad() {
```

Step 2

Add the private method `configureActivityItems()`, then create a mutable variable named `configuration` of type `UIActivityItemsConfiguration`.

`configuration` will contain the activity items configuration for the selected recipe if the view controller has one. Otherwise, `configuration` will be `nil`.

Step 3

If the view controller has a recipe, set `configuration` to a new instance of `UIActivityItemsConfiguration`, passing in the recipe's `fullImage`. Then assign a closure to the configuration's `metaProvider`, setting the title and message body equal to the recipe's `title`.

Note

`metaProvider` contains additional recipe information that the sharing service can include when sharing the recipe. For example, a recipe the user shares through Messages includes the recipe title and a photo.

Step 4

Add a `guard` statement to get a reference to the toolbar delegate as a type of `ToolbarDelegate`.

Note

The `titlebar` property of `UIWindowScene` is only available to apps built with Mac Catalyst, so enclose the code that retrieves the toolbar delegate in a `targetEnvironment(macCatalyst)` conditional block.

Step 5

If the toolbar delegate has a reference to a Share Recipe toolbar item, set the item's `activityItemsConfiguration` property to the `UIActivityItemsConfiguration` created in Step 3.

`RecipeDetailViewController` needs to call `configureActivityItems()` each time the user selects a recipe. The view controller also calls the method after the view appears.

Step 6

Scroll the editor to the top of `RecipeDetailViewController.swift`, and in `didSet` of the `recipe` property, add a call to `configureActivityItems()`.

Step 7

Override the method `viewDidAppear(_:)` and call `configureActivityItems()`.

Step 8

Build and run the app, and select a recipe. Click the Share Recipe toolbar item to see the list of share options.

Check Your Understanding

Question 1 of 3

The Share Recipe toolbar item is an instance of what class?

- `NSSharingServicePickerToolbarItem`
- `NSToolbarItem`
- `NSToolbarShareItem`

Submit | Next question

Next

Accessing Actions Using Menu Elements

In Chapter 3, you updated the Recipes app by replacing items from the iOS toolbar and navigation bars with equivalent items in a Mac-style toolbar. Two items still need to be replaced: New Recipe and Delete Recipe.

Get started

To submit feedback on documentation, visit Feedback Assistant.

Light Dark Auto