

Interface Refinements

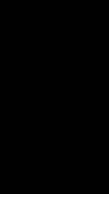
Supporting the Touch Bar

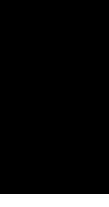
The Touch Bar is an input device located above the keyboard on supported MacBook Pro models. People can use the dynamic controls in the Touch Bar to interact with content on the main screen and gain quick access to system-level and app-specific functionality based on the current context.

In this tutorial, you'll add items to the Touch Bar so users can edit and delete a recipe, and designate a recipe as a favorite.

Download the project files and follow these steps to add Touch Bar support to the Recipes app.

10minsEstimated Time

Project files [@](#)

Xcode 12.2 or later >



Section 1

Add Controls to the Touch Bar

Provide users the option to interact with the Recipes app using the Touch Bar.

Important

If your Mac doesn't support Touch Bar, choose Window > Touch Bar > Show Touch Bar from the Xcode menu bar to use an onscreen Touch Bar to test your changes to the Recipes app.



Define identifiers for the delete recipe, edit recipe, and toggle favorite controls that will appear in the Touch Bar.

Step 1

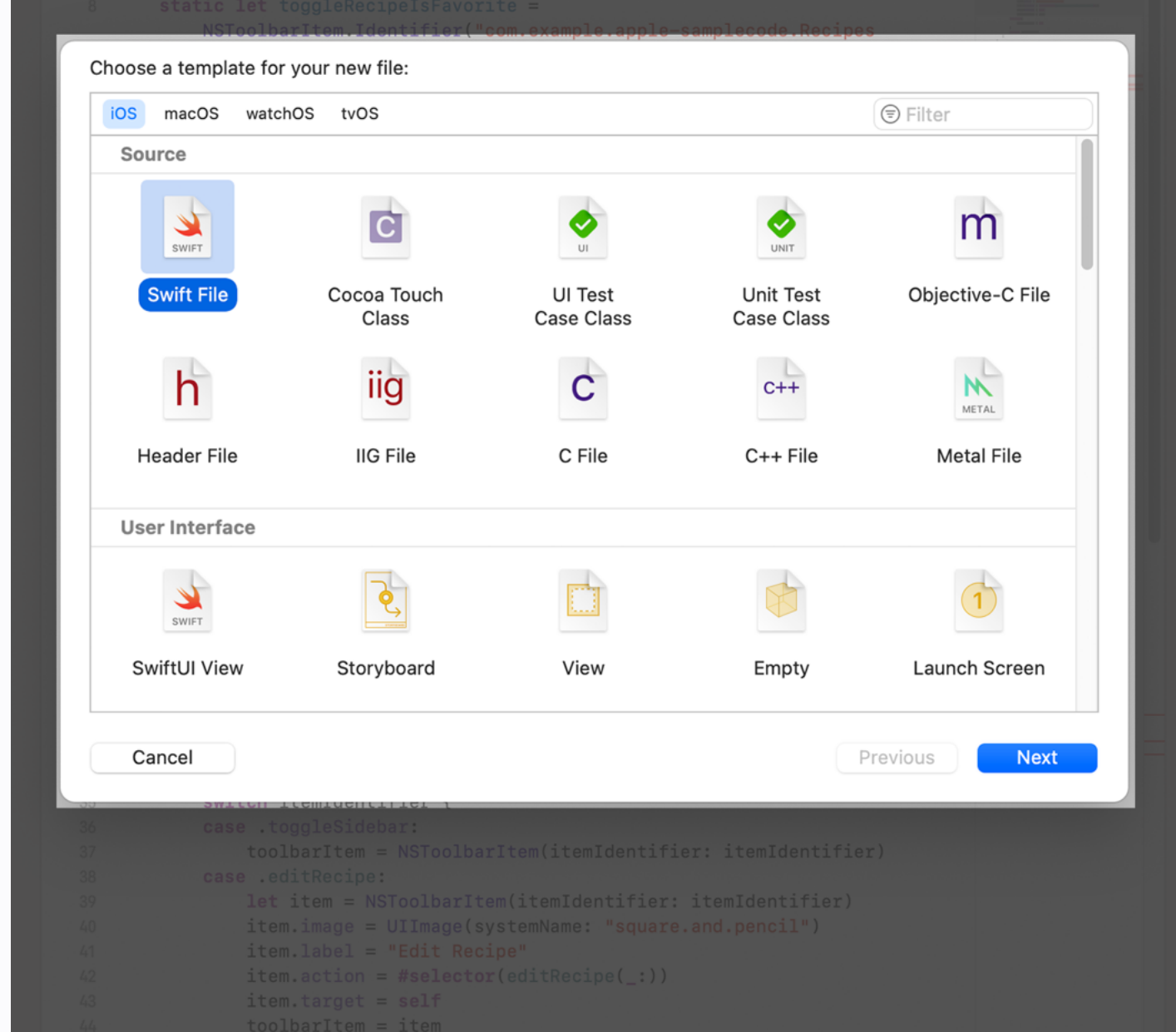
Choose File > New > File. Select the iOS platform and the Swift File template, and then click Next. Name the file `RecipeDetail TouchBar.swift` and click Create.

Step 2

Replace `import Foundation` with `import UIKit` and add a `targetEnvironment()` conditional block.

Note

The Touch Bar is only available in the Mac Catalyst target environment.



Step 3

Add an extension on `NSTouchBarItem.Identifier` inside the `targetEnvironment()` conditional block. Then add the identifiers `deleteRecipe`, `editRecipe`, and `toggleRecipeIsFavorite`.

Add the delete recipe, edit recipe, and toggle favorite controls to the Touch Bar.

Step 4

Add an extension on `RecipeDetailViewController` conforming to `NSTouchBarDelegate`.

Step 5

Override `makeTouchBar()` to create and configure an instance of `NSTouchBar`.

Step 6

Set the Touch Bar delegate to the current instance of `RecipeDetailViewController`.

Step 7

Set the default item identifiers for the Touch Bar, and return the Touch Bar instance.

Note

The Touch Bar displays items in the order they appear in the `defaultItemIdentifiers` array.

Provide the delete recipe, edit recipe, and toggle favorite item to the Touch Bar when asked.

Step 8

Add the Touch Bar delegate method `touchBar(_:makeItemForIdentifier:)` that contains a switch statement with a default case. Then return the Touch Bar item.

Note

The Touch Bar object calls this method to ask the delegate object for the Touch Bar item with the specified identifier.

Step 9

Add a `deleteRecipe` case to the switch statement.

Note

`RecipeDetailViewController` provides the action selectors to delete the recipe.

Step 10

Add an `editRecipe` case to the switch statement.

Note

`RecipeDetailViewController` provides the action selectors to edit the recipe.

Step 11

Add a `toggleRecipeIsFavorite` case to the switch statement.

Note

`RecipeDetailViewController` provides the action selectors to toggle the recipe's `isFavorite` property.

The Touch Bar now displays the controls from the Recipes app. However, the image displayed in `toggleRecipeIsFavorite` Touch Bar item doesn't change between a heart and filled heart when the recipe's `isFavorite` property changes. Let's fix that problem.

Step 12

In the Project navigator, select `RecipeDetailViewController.swift` to open the file in the editor. Then scroll to the `updateUI()` method and set the image for the toggle favorite Touch Bar item.

Experiment

Build and run the app, and test the new Touch Bar controls by selecting a recipe in the app. If your Mac doesn't support Touch Bar, choose Window > Touch Bar > Show Touch Bar from the Xcode menu bar to use an onscreen Touch Bar to test your changes to the Recipes app.

Check Your Understanding

Question 1 of 3

What is the Touch Bar?

The Touch Bar is an input view displayed above the virtual keyboard on iPad and iPhone.

The Touch Bar is an input device located above the keyboard on select MacBook Pro models.

The Touch Bar is an input window in macOS that displays dynamic controls people can use to interact with content in a main window of an app.

Submit

Next question

Next

Updating the App Icon

Recipes needs to provide icons in different sizes to make sure the app icon looks great on all supported devices, including icons sized specifically for the Mac.

Get started

Platforms

iOS

iPadOS

macOS

tvOS

visionOS

watchOS

Tools

Swift

SwiftUI

Swift Playgrounds

TestFlight

Xcode

Xcode Cloud

SF Symbols

Topics & Technologies

Accessibility

Accessories

App Extension

App Store

Audio & Video

Augmented Reality

Design

Distribution

Education

Fonts

Games

Health & Fitness

In-App Purchase

Localization

Maps & Location

Machine Learning

Open Source

Security

Safari & Web

Resources

Documentation

Tutorials

Downloads

Forums

Videos

Support

Support Articles

Contact Us

Bug Reporting

System Status

Account

Apple Developer

App Store Connect

Certificates, IDs, & Profiles

Feedback Assistant

Programs

Apple Developer Program

Apple Developer Enterprise Program

App Store Small Business Program

MFI Program

News Partner Program

Video Affiliate Program

Security Bounty Program

Security Research Device Program

Events

Meet with Apple

Apple Developer Centers

App Store Awards

Apple Design Awards

Apple Developer Academies

Entrepreneur Camp

WWDC