

[SwiftUI](#) / [Navigation](#) / Enhancing your app's content with tab navigation

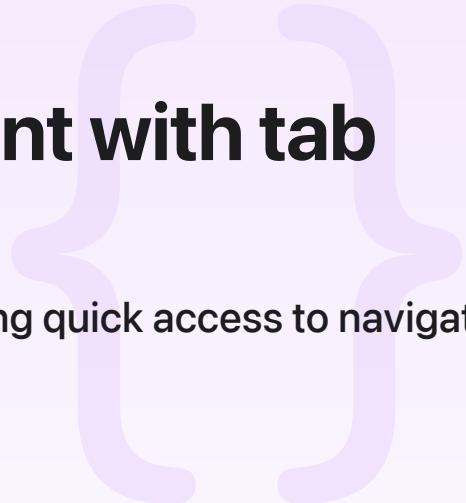
Sample Code

# Enhancing your app's content with tab navigation

Keep your app content front and center while providing quick access to navigation using the tab bar.

[Download \(1.2 GB\)](#)

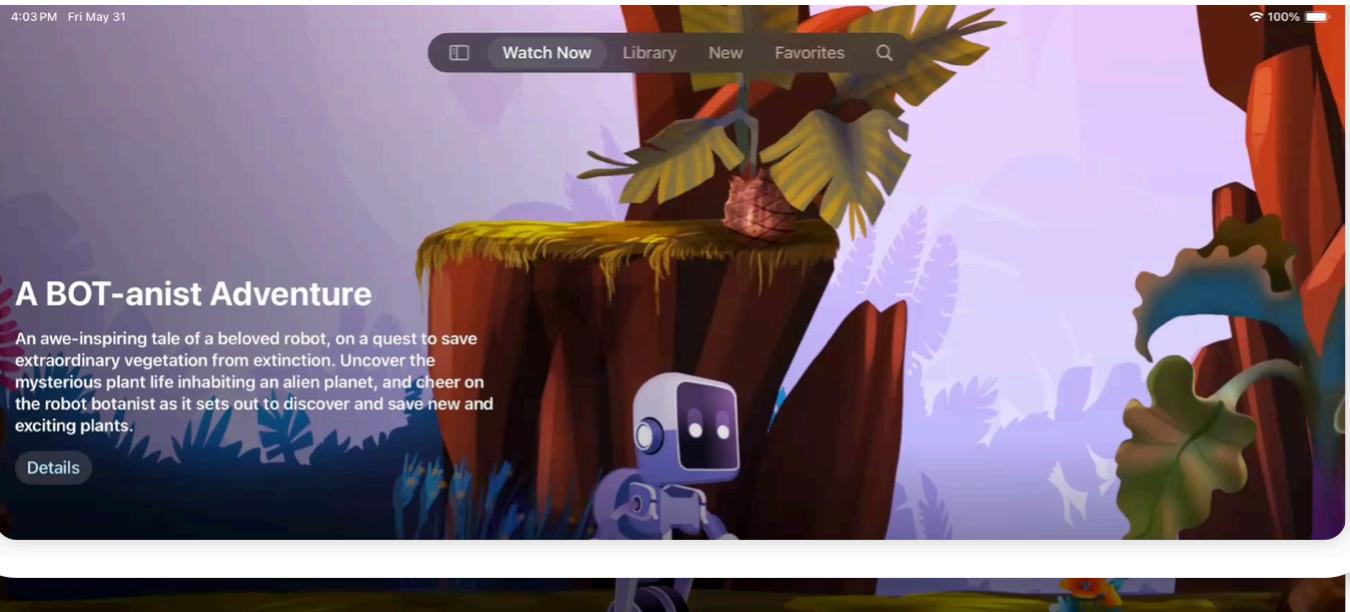
iOS 18.0+ | iPadOS 18.0+ | macOS 15.0+ | tvOS 18.0+ | visionOS 2.0+



## Overview

[Destination Video](#) adopts the `sidebarAdaptable` tab view style, which optimizes the content browsing experience for each platform.

Starting in iPadOS 18, the tab bar appears on the top of the screen floating over your content instead of appearing at the bottom of the screen. This appearance creates an immersive full-screen browsing experience. Tab bars provide people with access to the top-level navigation in your app. However, too many tabs can make it hard for people to locate content. Implementing a sidebar makes it easier to navigate a detailed information hierarchy.



Play ▶

## Create a tab bar

You can create a `TabView` with an explicit selection binding using the `init(selection:content:)` initializer. To add a tab within a `TabView` initialize a `Tab`. `DestinationVideo` uses the `init( :systemImage:value:content:)` initializer to create each tab:

```
@State private var selectedTab: Tabs = .watchNow

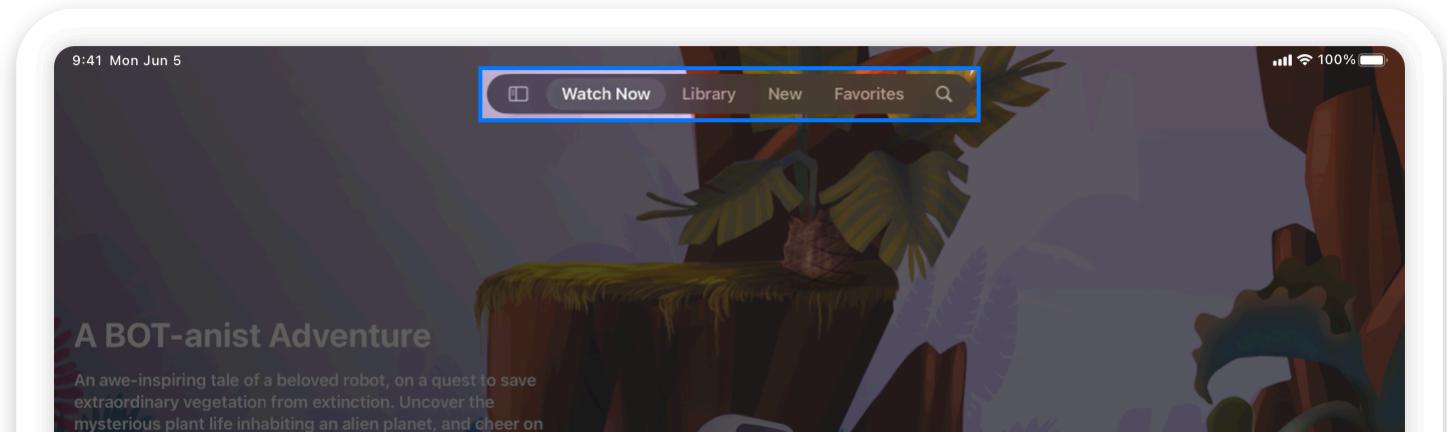
var body: some View {
    TabView(selection: $selectedTab) {
        Tab("Watch Now", systemImage: "play", value: .watchNow) {
            WatchNowView()
        }
        // More tabs...
    }
}
```

The selection value type of the `TabView` matches the value type of the tabs it contains. In this case, the value of each `Tab` is of type `Tabs`, which this sample defines the following enumeration:

```
enum Tabs: Equatable, Hashable, Identifiable {
    case watchNow
    case library
    case new
    case favorites
    case search
```

## Note

When using symbol images for your tabs, use the outline variant. The system automatically selects the filled variant when it appears in a tab bar.

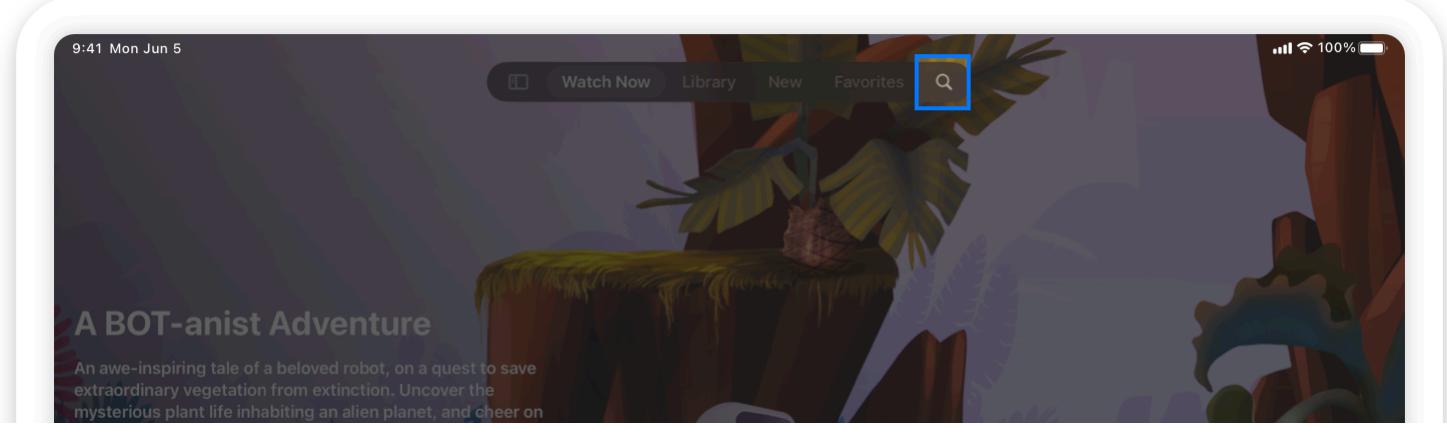


Additionally, this sample uses the `search` role with the `init(value:role:content:)` initializer. Setting the tab role to `search` makes the system applies a few default customizations to the Tab. The search tab gets:

- The default title for search, “search”
- The default system symbol for search, a magnifying glass
- The default pinned behavior for search, the system automatically pins it in the tab bar

```
Tab(value: .search, role: .search) {  
    // ...  
}
```

Pinned tabs appear at the trailing edge of the tab bar, depending on the preferred language of your app. When the language is a left-to-right language, they appear on the right side. When the language is a right-to-left language, they're on the left side.



# Build hierarchy in tab view

You can use a [TabSection](#) to declare a secondary tab hierarchy within a TabView. For example Destination Video uses the [init\(content:header:\)](#) initializer to create tab sections.

```
TabView(selection: $selectedTab) {  
    Tab("Watch Now", systemImage: "play", value: .watchNow) {  
        WatchNowView()  
    }  
  
    // More tabs...  
  
    TabSection {  
        Tab("Cinematic Shots", systemImage: "list.and.film", value: .collections(.ci  
            // ...  
        }  
    } header: {  
        Label("Collections", systemImage: "folder")  
    }  
}
```

Then it extends the Tabs enumeration to account for secondary tabs:

```
enum Tabs: Equatable, Hashable, Identifiable {  
    case watchNow  
    // ..  
    case search  
    case collections(Category)  
    case animations(Category)  
}  
  
enum Category: Equatable, Hashable, Identifiable, CaseIterable {  
    case cinematic  
    case forest  
    case sea  
    // ...  
}
```

This sample uses a [ForEach](#) loop to iterate and initialize a new Tab for each tab value.

```
TabSection {
    ForEach(Category.collectionsList) { collection in
        Tab(collection.name, systemImage: collection.icon, value: Tabs.collections[collection])
            // ...
    }
}

} header: {
    Label("Collections", systemImage: "folder")
}
```

## Make the tab bar adaptable

Tab bars with the `sidebarAdaptable` style allow people to toggle between the sidebar and tab bar. This lets your app leverage the convenience of being able to quickly navigate to top-level destinations within a compact tab bar while providing rich navigation hierarchy and destination options in the sidebar.

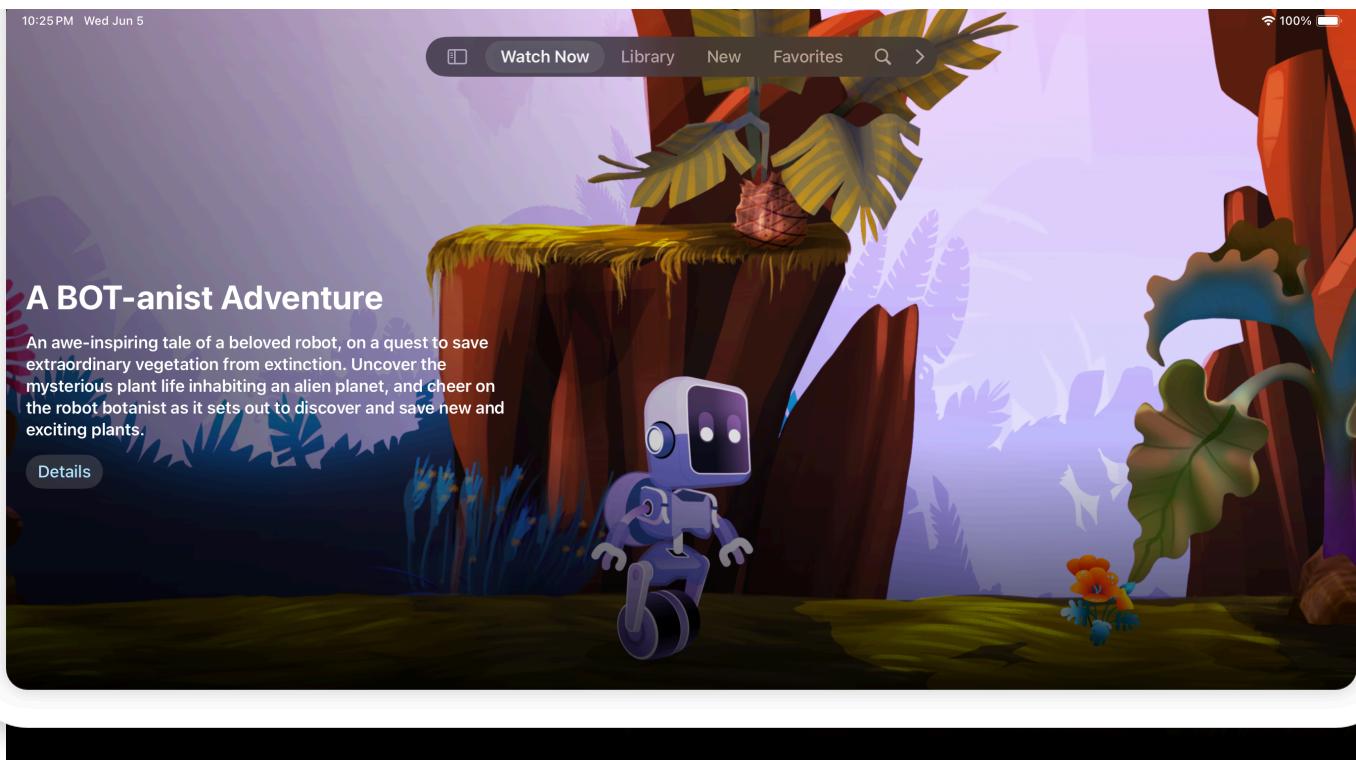
To create an adaptable tab bar, Destination Video adds the `tabViewStyle(_ :)` modifier to its `TabView` and passes in the value `sidebarAdaptable`.

```
TabView(selection: $selectedTab) {
    // Tabs
    // ...
}

.tabViewStyle(.sidebarAdaptable)
```

A `TabView` with the `sidebarAdaptable` style appears differently depending on the platform, as shown in the following images.

iPadOS    iOS    macOS    tvOS    visionOS



Play 

A TabView appears as top tab bar that becomes a sidebar in iPadOS. Tab sections appear in both the sidebar and the tab bar. In compact view, a TabView appears as a bottom tab bar.

### Note

By default, contents in a ScrollView(.horizontal) scroll under the sidebar when you use the sidebarAdaptable tab view style in iPadOS. You can prevent the content from scrolling under the sidebar by adding the `clipped(antialiased:)` or `clipShape(_:style:)` modifier to ScrollView.

## Enable customization

Tab view customization allows people to enter edit mode and personalize the tab bar. The customization in Destination Video allows people to:

- Drag and drop tabs to remove and add tabs to the tab bar
- Hide non-essential tabs
- Reorder tabs in tab sections in the sidebar
- Reorder tabs in the tab bar

To enable customizations, this sample defines a `TabViewCustomization` and attaches it to the TabView using the `tabViewCustomization(_:)` modifier. To persist the customization, this sample adds `AppStorage` with an identifier for a `TabViewCustomization` variable. Finally, it adds the `customizationID(_:)` modifier to each tab.

```
@AppStorage("sidebarCustomizations") var tabViewCustomization: TabViewCustomization
@State private var selectedTab: Tabs = .watchNow

var body: some View {
    TabView(selection: $selectedTab) {
        Tab("Watch Now", systemImage: "play", value: .watchNow) {
            WatchNowView()
        }
        .customizationID(Tabs.watchNow.customizationID)

        // More tabs...
    }
    .tabViewCustomization($tabViewCustomization)
}
```

To keep the most important tabs visible and in a fixed position, turn off customization behavior for those tabs using the `customizationBehavior(_:_for:)` modifier.

```
Tab("Watch Now", systemImage: "play", value: .watchNow) {
    WatchNowView()
}
.customizationBehavior(.disabled, for: .sidebar, .tabBar)
```

## Set the default visibility for tabs

In iPadOS, if there are too many tabs to fit in the screen, the system collapses the tabs that don't fit and enables scrolling. However, having too many tabs can make it harder for people to locate the tab they're looking for and navigate your app. Consider limiting the number of tabs so they all fit in the tab bar. The `defaultVisibility(_ :for:)` modifier sets the default visibility of a Tab or TabSection.

Destination Video contains five tabs and two tab sections, each tab section contains multiple secondary tabs, but only seven tabs appear in the tab bar. In order to limit the tab bar to the most important tabs, all tabs within a TabSection are hidden from the tab bar by default.

```
TabSection {  
    // Tabs  
} header {  
    // Section header  
}  
.defaultVisibility(.hidden, for: .tabBar)
```

iPadOS    iOS    macOS    tvOS    visionOS



Watch Now

Library

New

Favorites



## A BOT-anist Adventure

An awe-inspiring tale of a beloved robot, on a quest to save extraordinary vegetation from extinction. Uncover the mysterious plant life inhabiting an alien planet, and cheer on

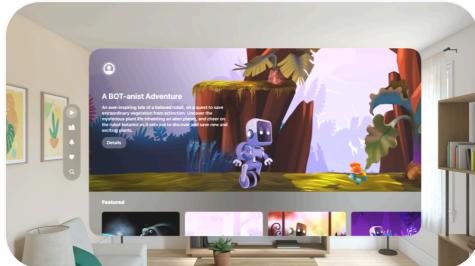
[Details](#)[Play](#) 

If you enable customization, the `defaultVisibility(_:for:)` modifier still allows people to drag a tab from the sidebar into the tab bar. If you want to restrict tabs to only appear in the sidebar use `sidebarOnly` instead of setting the default visibility.

For design guidance, see Human Interface Guidelines > [Tab bars](#).

## See Also

### Related samples



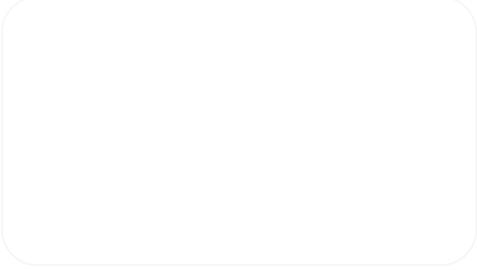
### Destination Video

### Related articles

- Elevating your iPad app with a tab bar and sidebar

Provide a compact, ergonomic tab bar for quick access to key parts of your app, and a sidebar for in-depth navigation.

### Related videos



**Elevate your tab and sidebar  
experience in iPadOS**