

[UIKit](#) / [UITabBar](#)

Class

UITabBar

A control that displays one or more buttons in a tab bar for selecting between different subtasks, views, or modes in an app.

iOS 2.0+ | iPadOS 2.0+ | Mac Catalyst 13.1+ | tvOS | visionOS 1.0+

```
@MainActor
class UITabBar
```

Overview

Typically, you use tab bars in conjunction with a [UITabBarController](#) object, but you can also use them as standalone controls in your app. Tab bars always appear across the bottom edge of the screen and display the contents of one or more [UITabBarItem](#) objects. A tab bar's appearance can be customized with a background image or tint color to suit the needs of your interface. Tapping an item selects and highlights that item, and you use the selection of the item to enable the corresponding mode for your app.

You can configure tab bars programmatically or in Interface Builder. A [UITabBarController](#) object provides its own tab bar object and you must configure the object provided to you. When creating a tab bar programmatically, use the [init\(frame:\)](#) method or another view initializer method to set its initial configuration. Use the methods of this class to configure the appearance of the tab bar. For tab bars you create yourself, you also use the methods of this class to specify the items displayed by the tab bar.

Note

The [UITabBar](#) class and [UIToolbar](#) classes have similar appearances but different purposes. Use tab bars to convey and change your app's mode. Use toolbars to present the user with a set of actions that are relevant to the currently presented content.

A tab bar reports selections and user customizations to its delegate object. For tab bars you create yourself, use the delegate to respond to selections or to the addition, removal, or reordering of items in the tab bar. (A [UITabBarController](#) object acts as the delegate for the tab bar it manages.) For more information on implementing a tab bar delegate, see [UITabBarDelegate](#).

Configure the tab bar items

You can configure tab bar items using Interface Builder or create and configure them programmatically in your code. Tab bars in Interface Builder come preconfigured with some initial items and you can add, remove, or reorder items as needed. How you configure items at design time depends on whether your tab bar is associated with a [UITabBarController](#) object:

- Configuring your tab bar in Interface Builder:
- When a [UITabBarController](#) object is present, add or remove view controllers to your scene and create relationship segues between the tab bar controller and each new view controller. Creating a relationship segue automatically adds a new item to the tab bar, and deleting an existing relationship segue removes the corresponding tab bar item.
- When a tab bar controller isn't present, drag tab bar items from the library onto your tab bar.
- Configuring your tab bar programmatically:
- To configure the tab bar associated with a [UITabBarController](#) object, configure the view controllers associated with the tab bar controller. The tab bar automatically obtains its items from the [tabBarItem](#) property of each view controller associated with the tab bar controller.
- To configure tab bar items directly, use the [setItems\(_:animated:\)](#) method of the tab bar itself.

A tab bar displays all of its tabs onscreen at once, using the [itemPositioning](#) property to determine how to position items in the available space. If you have more items than can fit in the available space, display only a subset of them and let the user select which tabs are displayed. The [beginCustomizingItems\(_:\)](#) method displays an interface for selecting which tab bar items to display.

The contents of each item are stored in a [UITabBarItem](#) object. Each item contains a title and an image to display in the tab. You can also use tab bar items to add a badge to the corresponding tab. For more information about creating and configuring items, see [UITabBarItem](#).

Respond to tab selections

For tab bars with an associated tab bar controller, the tab bar controller automatically manages selections and displays the appropriate view controller. The only time you have to manage selections yourself is when you create the tab bar without a tab bar controller. The tab bar reports selections to the `tabBar(_:didSelect:)` method of its `delegate` object, which you can use to respond to selection changes. For more information about implementing the delegate object, see [UITabBarDelegate](#).

Configure a tab bar with Interface Builder

The following table lists the attributes that you configure for tab bars in Interface Builder.

Attribute	Discussion
Background	The background image to display for the bar. If you specify a stretchable image, the image is stretched to fit the available space; otherwise, the image is tiled. When you configure a background image, the tab bar ignores the tint color information. To set this attribute programmatically, use the backgroundImage property.
Shadow	The custom shadow image for the tab bar. This attribute is ignored if the tab bar does not also have a custom background image. To set this attribute programmatically, use the shadowImage property.
Selection	The image to use for the selected tab. To set this attribute programmatically, use the selectionIndicatorImage property.
Image Tint	The tint color to apply to the selected item. To set this attribute programmatically, use the tintColor property.
Style	The basic style to apply to the bar. You can configure a tab bar with a dark or light style and the bar can be opaque or translucent. To set the style programmatically, use the barStyle and isTranslucent properties.
Bar Tint	The tint color to apply to the bar. To set this attribute programmatically, use the barTintColor property.
Item Positioning	The positioning scheme to apply to items. Use this attribute to configure how items are spaced across the length of the tab bar. To set this attribute programmatically, use the itemPositioning property.

Internationalize a tab bar

To internationalize a tab bar, you must provide localized strings for the tab bar item titles.

For more information, see [Localization](#).

Make a tab bar accessible

Tab bars are accessible by default.

With VoiceOver enabled on an iOS device, when a user touches a tab in a tab bar, VoiceOver reads the title of the tab, its position in the bar, and whether it's selected. For example in the iTunes app on iPad, you might hear "Selected, Audiobooks, four of seven" or "Genius, six of seven."

For general information about making your interface accessible, see [Accessibility for UIKit](#).

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

Topics

Customizing the tab bar behavior

```
var delegate: (any UITabBarDelegate)?
```

The tab bar's delegate object.

```
protocol UITabBarDelegate
```

The [UITabBarDelegate](#) protocol defines optional methods for a delegate of a [UITabBar](#) object. The [UITabBar](#) class provides the ability for the user to reorder, remove, and add items to the tab bar; this process is referred to as customizing the tab bar. The tab bar delegate receives messages when customizing occurs.

Configuring tab bar items

```
var items: [UITabBarItem]?
```

The items displayed by the tab bar.

```
func setItems([UITabBarItem]?, animated: Bool)
```

Sets the items on the tab bar, optionally animating any changes into position.

```
var selectedItem: UITabBarItem?
```

The currently selected item on the tab bar.

Customizing tab bar appearance

`var standardAppearance: UITabBarAppearance`

The appearance settings for a standard-height tab bar.

`var scrollEdgeAppearance: UITabBarAppearance?`

The appearance settings for the tab bar when the edge of scrollable content aligns with the edge of the tab bar.

`var leadingAccessoryView: UIView`

The view at the leading edge of a tab bar on tvOS.

`var trailingAccessoryView: UIView`

The view at the trailing edge of a tab bar on tvOS.

`var isTranslucent: Bool`

A Boolean value that indicates whether the tab bar is translucent.

 Legacy customizations

Customize appearance information directly on the tab bar object.

Supporting user customization of tab bars

`func beginCustomizingItems([UITabBarItem])`

Presents a standard interface that lets the user customize the contents of the tab bar.

`func endCustomizing(animated: Bool) -> Bool`

Dismisses the standard interface used to customize the tab bar.

`var isCustomizing: Bool`

A Boolean value indicating whether the user is currently customizing the tab bar.

Relationships

Inherits From

UIView

Conforms To

CALayerDelegate
CVarArg
Copyable
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSCoding
NSObjectProtocol
NSTouchBarProvider
Sendable
SendableMetatype
UIAccessibilityIdentification
UIActivityItemsConfigurationProviding
UIAppearance
UIAppearanceContainer
UICoordinateSpace
UIDynamicItem
UIFocusEnvironment
UIFocusItem
UIFocusItemContainer
UILargeContentViewerItem
UIPasteConfigurationSupporting
UIPopoverPresentationControllerSourceItem
UIResponderStandardEditActions
UISpringLoadedInteractionSupporting
UITraitChangeObservable
UITraitEnvironment
UIUserActivityRestoring

See Also

Container view controllers

 [Creating a custom container view controller](#)

Create a composite interface by combining content from one or more view controllers with other custom views.

`class UISplitViewController`

A container view controller that implements a hierarchical interface.

`class UINavigationController`

A container view controller that defines a stack-based scheme for navigating hierarchical content.

`class UINavigationController`

Navigation controls that display in a bar along the top of the screen, usually in conjunction with a navigation controller.

`class UINavigationController`

The items that a navigation bar displays when the associated view controller is visible.

`class UITabBarController`

A container view controller that manages a multiselection interface, where the selection determines which child view controller to display.

`class UITabBarItem`

An object that describes an item in a tab bar.

`class UITab`

An object that manages a tab in a tab bar.

`class UITabAccessory`

`class UISearchTab`

A tab subclass that represents the system's search tab.

`class UITabGroup`

An object that manages a collection of tab objects.

`class UIPageViewController`

A container view controller that manages navigation between pages of content, where a subview controller manages each page.