

All Technologies

SwiftUI

WidgetConfiguration

Implementing a widget

var body: Self.Body

Body

Setting a name

func configurationDisplayName(...)

Setting a description

func description(\_:)

Setting the appearance

func supportedFamilies([Widget...)

func contentMarginsDisabled() -...

func disfavoredLocations([Widg...)

func containerBackgroundRemo...

Managing background tasks

func backgroundTask<D, R>(Bac...

func onBackgroundURLSessionE...

Instance Methods

func associatedKind(String?) -> ...

Filter

/

SwiftUI / WidgetConfiguration

Protocol

WidgetConfiguration

A type that describes a widget's content.

iOS 14.0+ | iPadOS 14.0+ | Mac Catalyst 14.0+ | macOS 11.0+ | visionOS 1.0+ | watchOS 9.0+

@MainActor @preconcurrency protocol WidgetConfiguration

Overview

A type conforming to this protocol inherits @preconcurrency @MainActor isolation from the protocol if the conformance is included in the type's base declaration:

struct MyCustomType: Transition { // `@preconcurrency @MainActor` isolation by default }

Isolation to the main actor is the default, but it's not required. Declare the conformance in an extension to opt out of main actor isolation:

extension MyCustomType: Transition { // `nonisolated` by default }

Topics

Implementing a widget

var body: Self.Body  
The content and behavior of this widget.  
Required  
associatedtype Body : WidgetConfiguration  
The type of widget configuration representing the body of this configuration.  
Required

Setting a name

func configurationDisplayName(\_:)  
Sets the localized name shown for a widget when a user adds or edits the widget.

Setting a description

func description(\_:)  
Sets the description shown for a widget when a user adds or edits it using the contents of a text view.

Setting the appearance

func supportedFamilies([WidgetFamily]) -> some Widget Configuration  
Sets the sizes that a widget supports.  
func contentMarginsDisabled() -> some WidgetConfiguration  
Disable default content margins.  
func disfavoredLocations([WidgetLocation], for: [Widget Family]) -> some WidgetConfiguration  
Sets the disfavored locations for a widget.  
func containerBackgroundRemovable(Bool) -> some Widget Configuration  
A modifier that marks the background of a widget as removable.

Managing background tasks

func backgroundTask<D, R>(BackgroundTask<D, R>, action: (D) async -> R) -> some WidgetConfiguration  
Runs the given action when the system provides a background task.  
func onBackgroundURLSessionEvents(matching:\_:)  
Adds an action to perform when events related to a URL session identified by a closure are waiting to be processed.

Instance Methods

func associatedKind(String?) -> some WidgetConfiguration  
Tells the system that a relevance-based widget can replace a timeline-based widget.  
func promptsForUserConfiguration() -> some Widget Configuration  
Specifies that a widget's configuration UI should be automatically presented after the widget is added.  
func pushHandler(any WidgetPushHandler.Type) -> some Widget Configuration  
Register a type that can handle push tokens changing for widgets.  
func supplementalActivityFamilies([ActivityFamily]) -> some WidgetConfiguration  
Sets the sizes that a Live Activity supports.  
func supportedMountingStyles([WidgetMountingStyle]) -> some WidgetConfiguration  
Specifies the mounting style for this widget.  
func widgetTexture(WidgetTexture) -> some Widget Configuration  
Specifies the widget texture for this widget.

Relationships

Conforming Types

EmptyWidgetConfiguration, LimitedAvailabilityConfiguration

See Also

Creating widgets

- Building Widgets Using WidgetKit and SwiftUI  
Create widgets to show your app's content on the Home screen, with custom intents for user-customizable settings.
  - Creating a widget extension  
Display your app's content in a convenient, informative widget on various devices.
  - Keeping a widget up to date  
Plan your widget's timeline to show timely, relevant information using dynamic views, and update the timeline when things change.
  - Making a configurable widget  
Give people the option to customize their widgets by adding a custom app intent to your project.
- protocol Widget  
The configuration and content of a widget to display on the Home screen or in Notification Center.
- protocol WidgetBundle  
A container used to expose multiple widgets from a single widget extension.
- struct LimitedAvailabilityConfiguration  
A type-erased widget configuration.
- struct EmptyWidgetConfiguration  
An empty widget configuration.

Platforms

- iOS
- iPadOS
- macOS
- tvOS
- visionOS
- watchOS

Tools

- Swift
- SwiftUI
- Swift Playground
- 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 & AI
- 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 Partner Program
- Security Bounty Program
- Security Research Device Program
- Events
- Meet with Apple
- Apple Developer Centers
- Apple Store Awards
- Apple Design Awards
- Apple Developer Academies
- WWDC