SwiftUI / View / translationTask(source:target:action:)

Instance Method

translationTask(source:target:action:)

Adds a task to perform before this view appears or when the specified source or target languages change.

Translation | SwiftUI | iOS 18.0+ | iPadOS 18.0+ | Mac Catalyst 26.0+ | macOS 15.0+

```
nonisolated
func translationTask(
    source: Locale.Language? = nil,
    target: Locale.Language? = nil,
    action: @escaping (TranslationSession) async -> Void
) -> some View
```

Parameters

`source`

The language the source content is in. If this is `nil`, the session tries to identify the language and prompts the person to pick the source language if it’s unclear. All text translated within the session should be in the same source language. Changing this value cancels previous tasks and creates a new session to perform translations again.

`target`

The language to translate content into. A `nil` value means the session picks a target according to the person’s `Locale.preferredLanguages` and the `source`. Changing this value cancels previous tasks and creates a new one to perform translations again.

`action`

A closure that runs when the view first appears and when `source` or `target` change. It provides a `TranslationSession` instance to perform one or multiple translations.

Discussion

This task provides an instance of `TranslationSession` to use to perform translations.

If you need to perform new translations again with the same `source` and `target` language, it’s better to use `translationTask(_:action:)` and call `invalidate()`.

For example, you can translate when content appears:

```swift
struct ContentView: View {
    var sourceText = "Hallo, Welt!"
    var sourceLanguage: Locale.Language?
    var targetLanguage: Locale.Language?

    @State private var targetText: String?

    var body: some View {
        Text(targetText ?? sourceText)
            .translationTask(
                source: sourceLanguage,
                target: targetLanguage
            ) { session in
                Task { @MainActor in
                    do {
                        let response = try await session.translate(sourceText)
                        targetText = response.targetText
                    } catch {
                        // code to handle error
                    }
                }
            }
    }
}
```

The system throws a `fatalError` if you use a `TranslationSession` instance after the attached view disappears or if you use it after changing the `source` or `target` parameters. This causes the `action` closure to provide a new `TranslationSession` instance.

See Also

Showing a translation

```
func translationPresentation(isPresented: Binding<Bool>, text: String, attachmentAnchor: PopoverAttachmentAnchor, arrowEdge: Edge, replacementAction: ((String) -> Void)?) -> some View
```

Presents a translation popover when a given condition is true.

```
func translationTask(TranslationSession.Configuration?, action: (TranslationSession) async -> Void) -> some View
```

Adds a task to perform before this view appears or when the translation configuration changes.