Swift Package Plugin In Use

Auto Generate your code

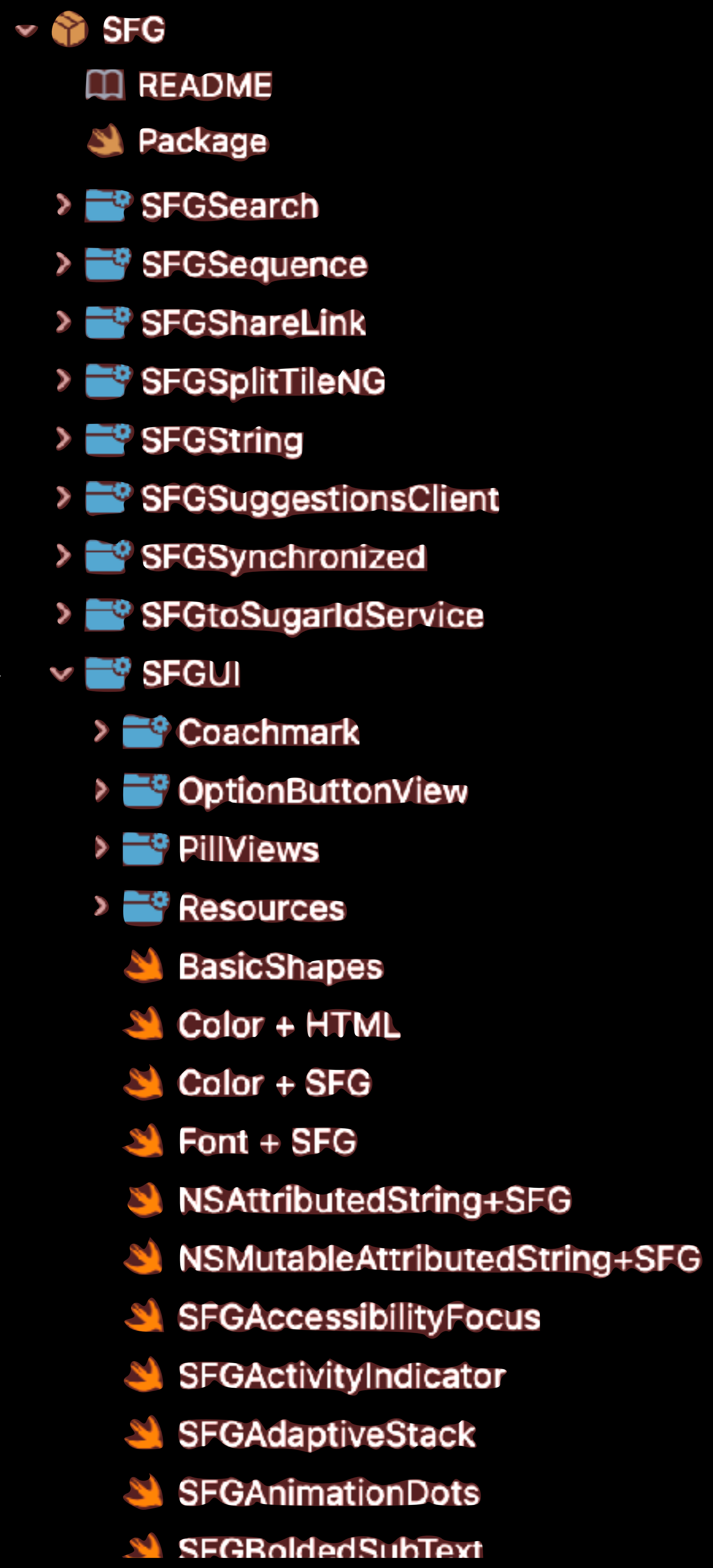
Shani Hajbi, iOS Staff Engineer at Shutterly

The problem

```
public extension Color {
   static var sfgFog:                 Color { .init(.sfgFog) }
   static var sfgFogLight:            Color { .init(.sfgFogLight) }
   static var sfgFogMedium:           Color { .init(.sfgFogMedium) }
   static var sfgDark:                Color { .init(.sfgDark) }
   static var sfgIgnite:              Color { .init(.sfgIgnite) }
```

```
public struct SFGAnimationDots: View {
    var circleWidth: CGFloat?
    var circleHeight: CGFloat?

    public var body: some View {
        HStack(alignment: .center, spacing: 6){
            ForEach([0.25, 0.5, 0.75], id:\.self) { delay in
                Dot(delay: delay,circleWidth: circleWidth, circleHeight: 
circleHeight)
            }

        }
    }
```

The problem



- SFG
  - README
  - Package
  - SFGSearch
  - SFGSequence
  - SFGShareLink
  - SFGSplitTileNG
  - SFGString
  - SFGSuggestionsClient
  - SFGSynchronized
  - SFGtoSugarIdService
  - SFGUI
    - Coachmark
    - OptionButtonView
    - PillViews
    - Resources
    - BasicShapes
    - Color + HTML
    - Color + SFG
    - Font + SFG
    - NSAttributedString+SFG
    - NSMutableAttributedString+SFG
    - SFGAccessibilityFocus
    - SFGActivityIndicator
    - SFGAdaptiveStack
    - SFGAnimationDots
    - SFGBoldedSubText

Zero Maintenance

Avoid changes in production code

```
public extension Color {
    static var sfgFog:            Color { .init(.sfgFog) }
    static var sfgFogLight:       Color { .init(.sfgFogLight) }
    static var sfgFogMedium:      Color { .init(.sfgFogMedium) }
    static var sfgDark:           Color { .init(.sfgDark) }
    static var sfgIgnite:         Color { .init(.sfgIgnite) }
```

WWDC 2022

```swift
Image(named: "Heart")
.resizable()
```

```swift
public let Heart = Image("Heart", bundle: .module)
```

```swift
Heart
.resizable()
```

BuildToolPlugin

```swift
import Foundation
import PackagePlugin


@main
struct SFGUIPreviews: BuildToolPlugin {
    func createBuildCommands(context: PackagePlugin.PluginContext, target: PackagePlugin.Target) async throws ->
[PackagePlugin.Command] {

        guard let target = target as? SourceModuleTarget else {
            return []
        }

        let paths = target.sourceFiles(withSuffix: "swift").map { $0.path }

        let output = context.pluginWorkDirectory.appending(["SFGPreviewsCollection.swift"])

        let allPaths = paths.map({ $0.string }).joined(separator: ",")

        return [.buildCommand(displayName: "Generating constants for SFGUI",
                             executable: try context.tool(named: "SFGUIPreviewsCollector").path,
                             arguments: [allPaths, output.string],
                             inputFiles: paths,
                             outputFiles: [output])]
    }
}
```

main.swift

```swift
var generatedCode = """
    import Foundation
    import SwiftUI


    struct SFGUIPreviewItem: Identifiable {
        let id = UUID().uuidString
        let previews: (any View)
        let title: String
        var readableTitle: String?
        var description: String?
    }

    struct SFGUIColorItem: Identifiable {
        let id = UUID().uuidString
        let title: String
        let color: Color
    }

    """
```

```swift
import Foundation
import SwiftUI


struct SFGUIPreviewItem: Identifiable {
    let id = UUID().uuidString
    let previews: (any View)
    let title: String
    var readableTitle: String?
    var description: String?
}

struct SFGUIColorItem: Identifiable {
    let id = UUID().uuidString
    let title: String
    let color: Color
}
```

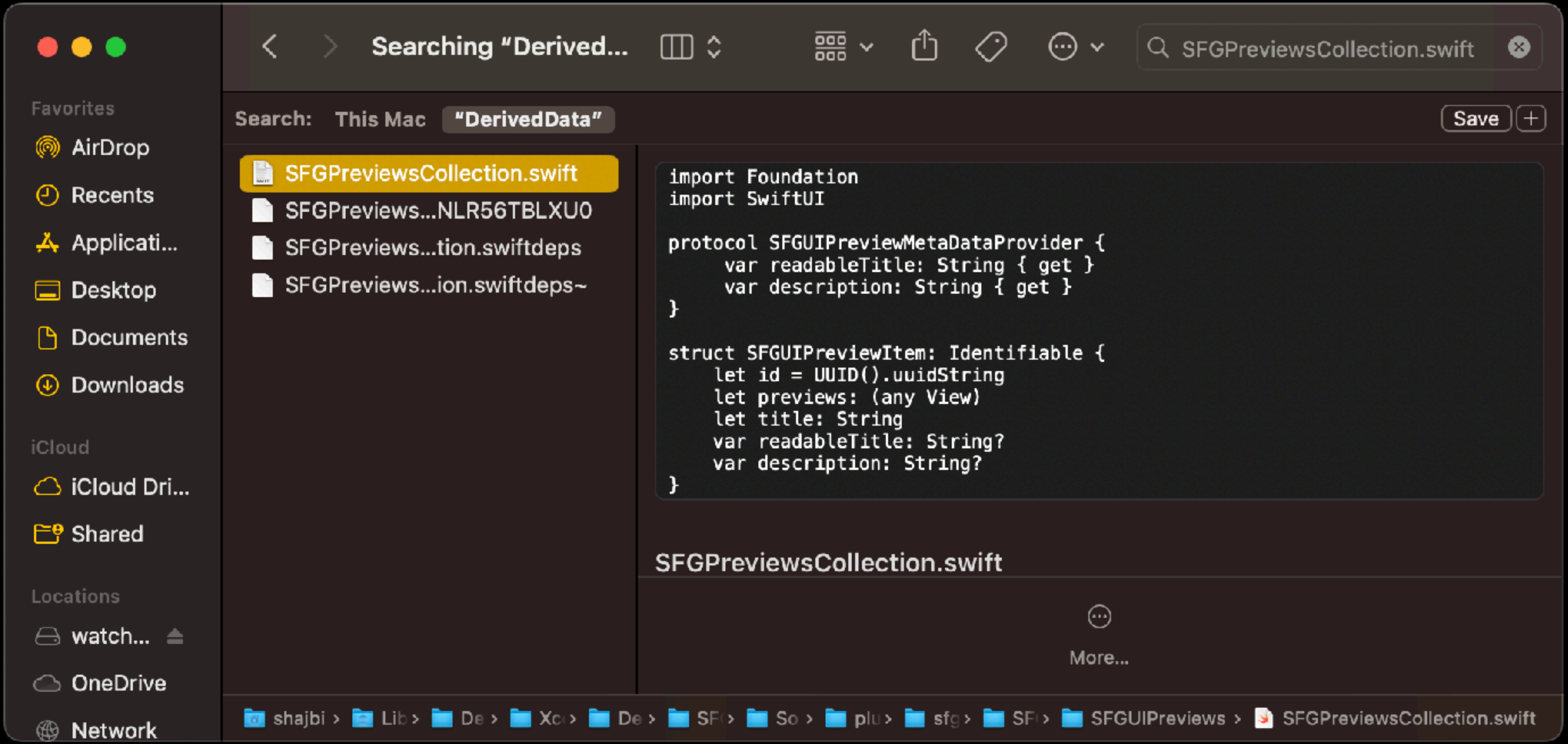
main.swift

```swift
regex.enumerateMatches(in: fileString, range: range) { (match, _, _) in
    if let matchRange = match?.range(at: 1),
       let swiftUIPreviewNameRange = Range(matchRange, in: fileString) {
        let previewName = fileString[swiftUIPreviewNameRange]

        print("Found preview: \(previewName)")


        let name = "\(previewName.lowercased())_previewItem"
        previewsGeneratedCode.append("""

            var \(name) = SFGUIPreviewItem(previews: \(previewName).previews,

                          title: "\(previewName.replacingOccurrences(of: "_Previews", with: ""))")

            let any_\(previewName): Any = \(previewName)()


            allPreviews.append(\(name))

        """)


    }
}
```

SFGPreviewsCollection.swift

```swift
func getAllSFGUIColors() -> [SFGUIColorItem] {

    var allColors: [SFGUIColorItem] = []


    let colorItem_sfgFog = SFGUIColorItem(title:  ".sfgFog", color: Color.sfgFog)
    allColors.append(colorItem_sfgFog)

    let colorItem_sfgFogLight = SFGUIColorItem(title:  ".sfgFogLight", color: Color.sfgFogLight)

    allColors.append(colorItem_sfgFogLight)




func getAllSFGUIPreviews() -> [SFGUIPreviewItem] {

  var allPreviews: [SFGUIPreviewItem] = []


  var sfgcoachmarkcontainerview_previews_previewItem =
      SFGUIPreviewItem(previews: SFGCoachMarkContainerView_Previews.previews, title: "SFGCoachMarkContainerView")


  allPreviews.append(sfgcoachmarkcontainerview_previews_previewItem)
```

SFGPreviewsCollection.swift
(Derived Data Folder)

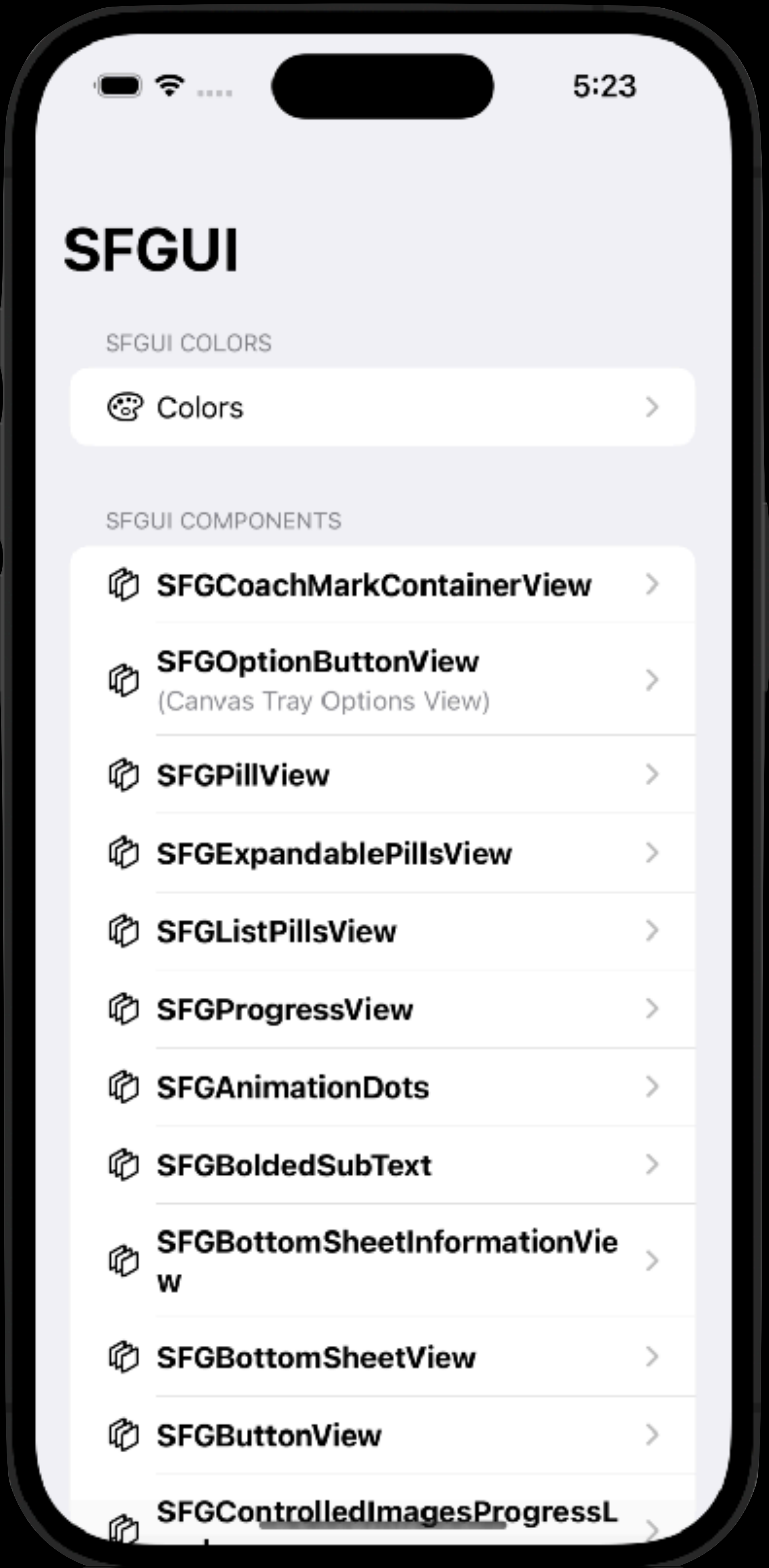


Searching "Derived...
SFGPreviewsCollection.swift
Favorites
AirDrop
Recents
Applicati...
Desktop
Documents
Downloads
iCloud
iCloud Dri...
Shared
Locations
watch...
OneDrive
Network
Search: This Mac "DerivedData"
Save
SFGPreviewsCollection.swift
SFGPreviews...NLR56TBLXU0
SFGPreviews...tion.swiftdeps
SFGPreviews...ion.swiftdeps~
SFGPreviewsCollection.swift
More...
shajbi › Lib › De › Xc › De › SF › So › plu › sfg › SF › SFGUIPreviews › SFGPreviewsCollection.swift

```
import Foundation
import SwiftUI

protocol SFGUIPreviewMetaDataProvider {
     var readableTitle: String { get }
     var description: String { get }
}

struct SFGUIPreviewItem: Identifiable {
    let id = UUID().uuidString
    let previews: (any View)
    let title: String
    var readableTitle: String?
    var description: String?
}
```

SFGPreviewsView.swift

```
@State var previews: [SFGUIPreviewItem] = getAllSFGUIPreviews()

Section(header: Text("SFGUI Components")) {
     ForEach(previews) { previewItem in
         SFGUIPreviewItemView(previewItem: previewItem)
     }
}
```


5:23

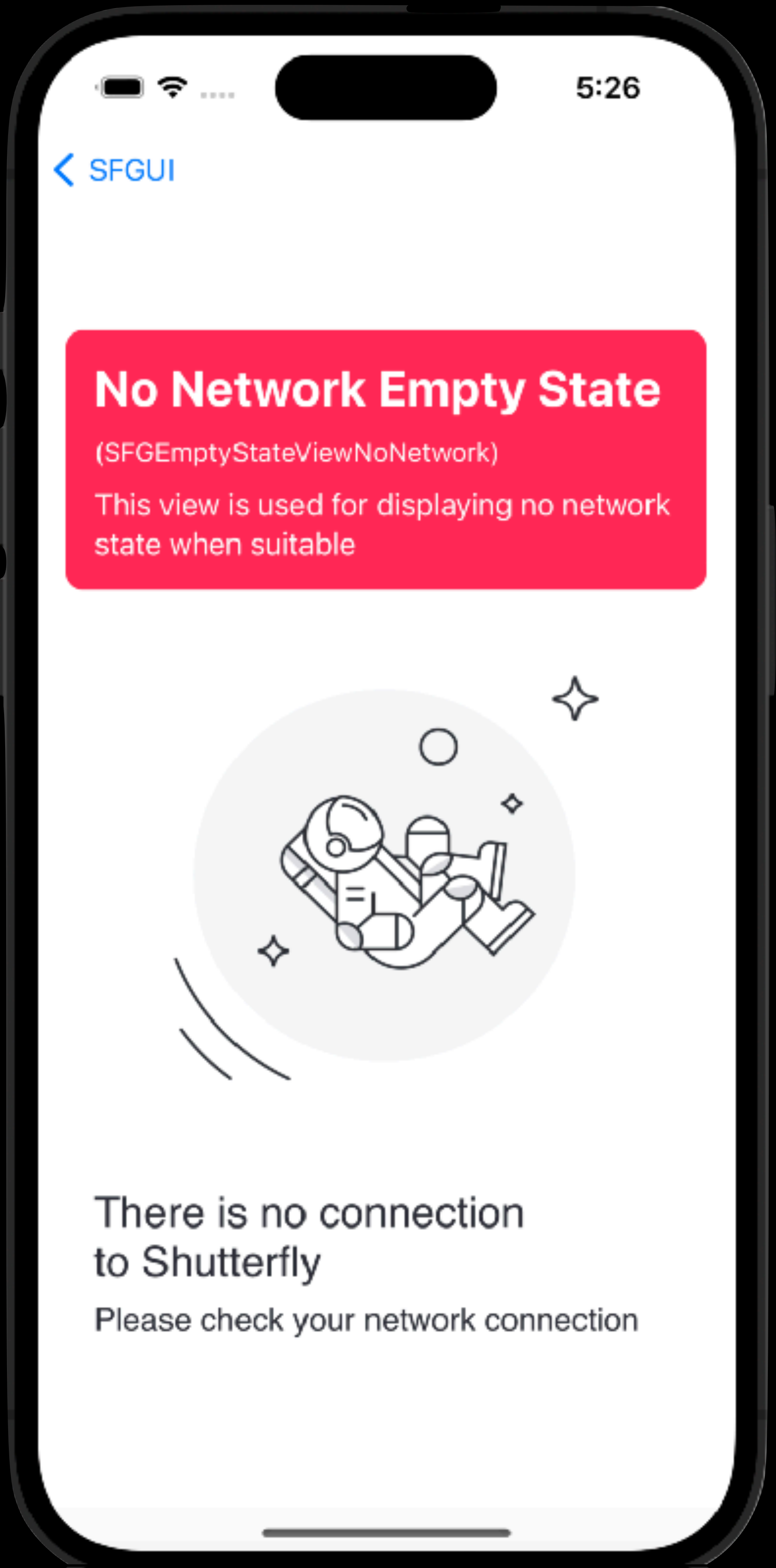
SFGUI

SFGUI COLORS

- Colors

SFGUI COMPONENTS

- SFGCoachMarkContainerView
- SFGOptionButtonView (Canvas Tray Options View)
- SFGPillView
- SFGExpandablePillsView
- SFGListPillsView
- SFGProgressView
- SFGAnimationDots
- SFGBoldedSubText
- SFGBottomSheetInformationView
- SFGBottomSheetView
- SFGButtonView
- SFGControlledImagesProgressL


5:26

SFGUI

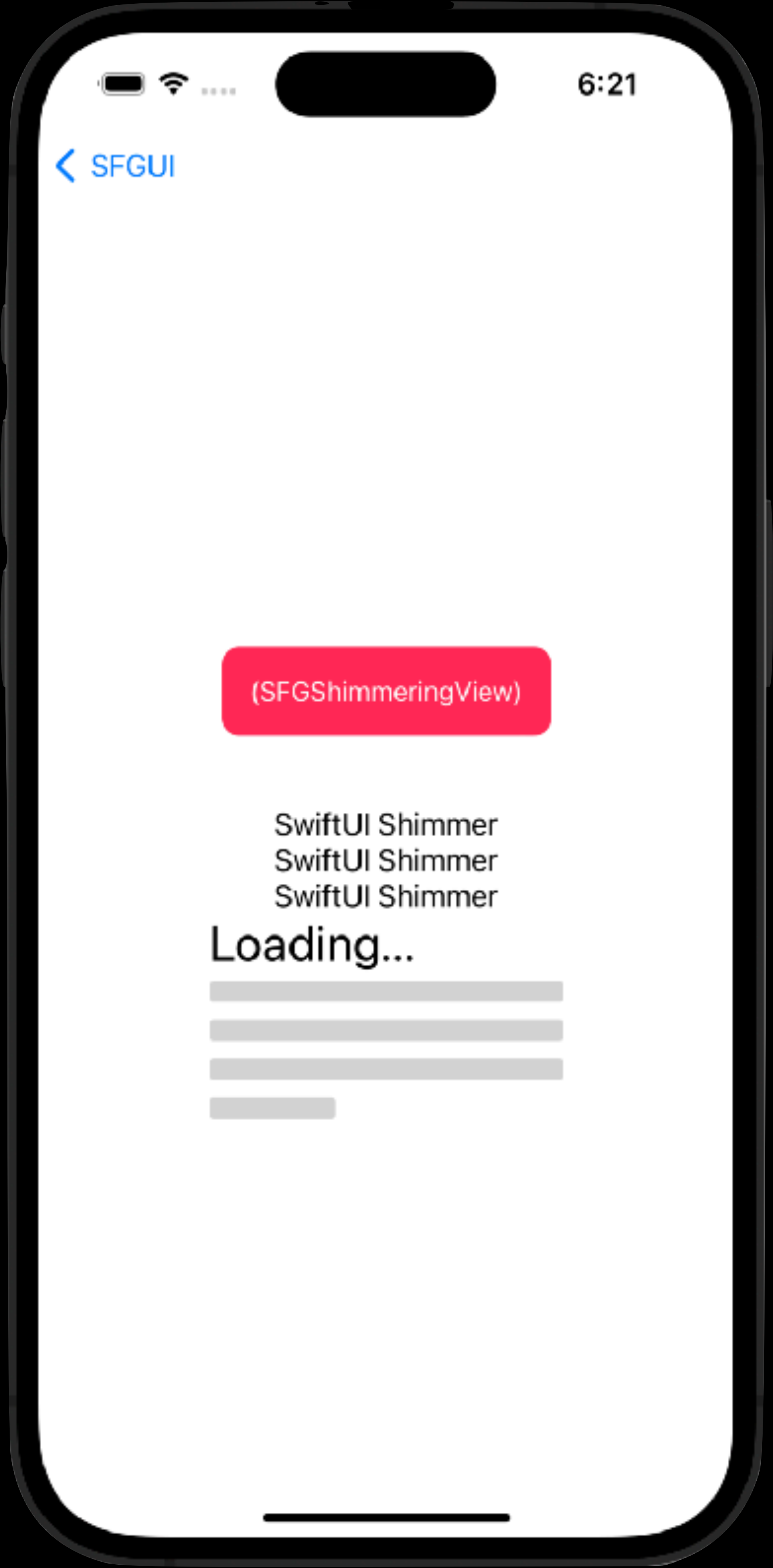
No Network Empty State

(SFGEmptyStateViewNoNetwork)

This view is used for displaying no network state when suitable

There is no connection to Shutterfly

Please check your network connection

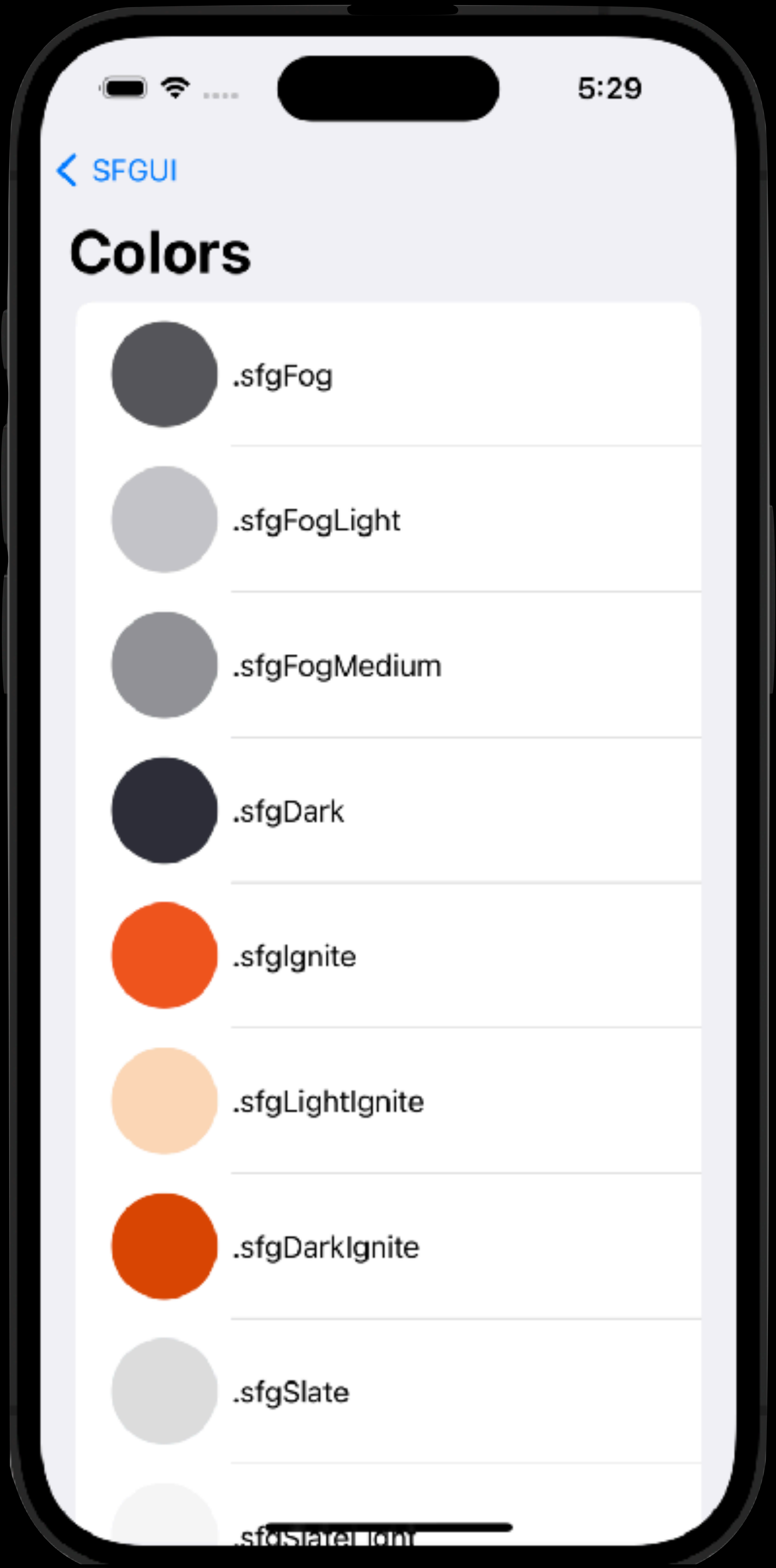

6:21

SFGUI

(SFGShimmeringView)

SwiftUI Shimmer
SwiftUI Shimmer
SwiftUI Shimmer

Loading...


5:29

SFGUI

Colors

- .sfgFog
- .sfgFogLight
- .sfgFogMedium
- .sfgDark
- .sfgIgnite
- .sfgLightIgnite
- .sfgDarkIgnite
- .sfgSlate
- .sfgSlateLight

Thanks