SwiftUI / Environment values

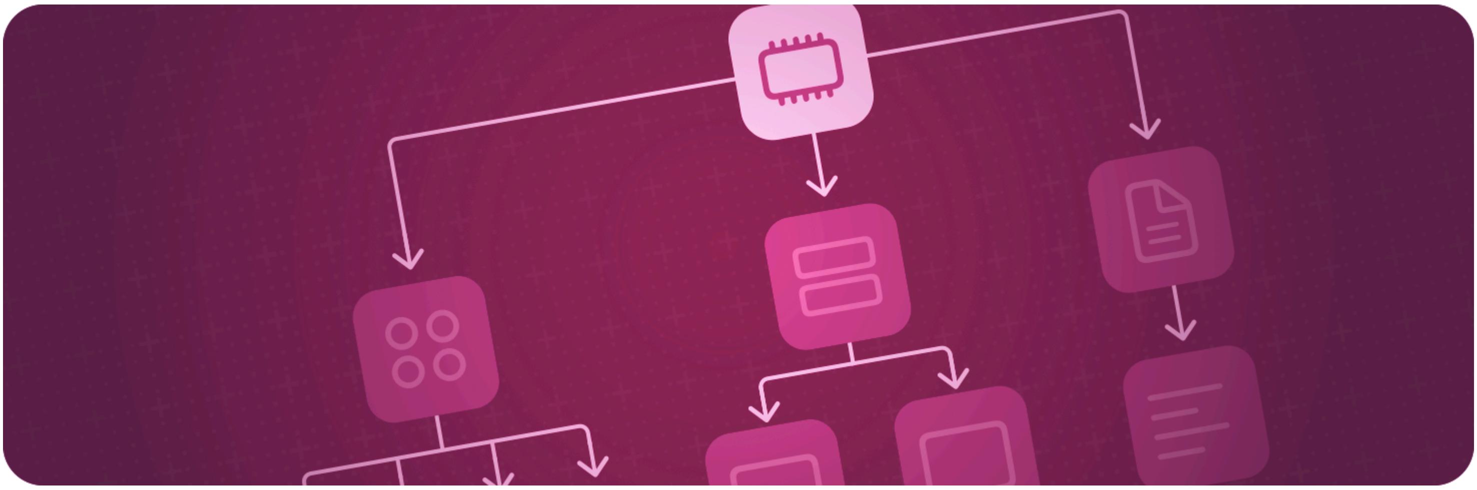
API Collection

Environment values

Share data throughout a view hierarchy using the environment.

Overview

Views in SwiftUI can react to configuration information that they read from the environment using an `Environment` property wrapper.



A view inherits its environment from its container view, subject to explicit changes from an `environment(_:_:)` view modifier, or by implicit changes from one of the many modifiers that operate on environment values. As a result, you can configure a entire hierarchy of views by modifying the environment of the group's container.

You can find many built-in environment values in the `EnvironmentValues` structure. You can also create a custom `EnvironmentValues` property by defining a new property in an extension to the environment values structure and applying the `Entry()` macro to the variable declaration.

Topics

Accessing environment values

```
struct Environment
```

A property wrapper that reads a value from a view's environment.

```
struct EnvironmentValues
```

A collection of environment values propagated through a view hierarchy.

Creating custom environment values

```
macro Entry()
```

Creates an environment values, transaction, container values, or focused values entry.

```
protocol EnvironmentKey
```

A key for accessing values in the environment.

Modifying the environment of a view

```
func environment<T>(T?) -> some View
```

Places an observable object in the view's environment.

```
func environment<V>(WritableKeyPath<EnvironmentValues, V>, V) -> some View
```

Sets the environment value of the specified key path to the given value.

```
func transformEnvironment<V>(WritableKeyPath<EnvironmentValues, V>, transform: (inout V) -> Void) -> some View
```

Transforms the environment value of the specified key path with the given function.

Modifying the environment of a scene

```
func environment<T>(T?) -> some Scene
```

Places an observable object in the scene's environment.

```
func environment<V>(WritableKeyPath<EnvironmentValues, V>, V) -> some Scene
```

Sets the environment value of the specified key path to the given value.

```
func transformEnvironment<V>(WritableKeyPath<EnvironmentValues, V>,
transform: (inout V) -> Void) -> some Scene
```

Transforms the environment value of the specified key path with the given function.

See Also

Data and storage

Model data

Manage the data that your app uses to drive its interface.

Preferences

Indicate configuration preferences from views to their container views.

Persistent storage

Store data for use across sessions of your app.