

[UIKit](#) /  / [UIImage](#) / Building high-performance lists and collection views

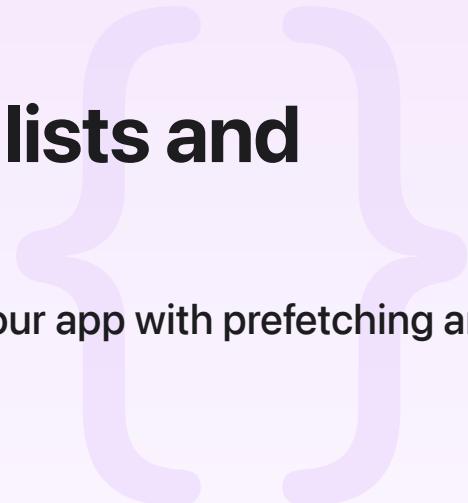
Sample Code

Building high-performance lists and collection views

Improve the performance of lists and collections in your app with prefetching and image preparation.

[Download](#)

iOS 15.0+ | iPadOS 15.0+ | Xcode 13.0+



Overview

Note

This sample code project is associated with WWDC21 session [10252: Make Blazing Fast Lists and Collection Views](#).

See Also

Loading and caching images

-  Providing images for different appearances
 - Supply image resources appropriate for light and dark appearances and for high-contrast environments.
-  Configuring and displaying symbol images in your UI

Create scalable images that integrate with your app's text, and adjust the appearance of those images dynamically.

Creating custom symbol images for your app

Create, organize, and annotate symbol images using SF Symbols.

`init?(named: String, in: Bundle?, compatibleWith: UITraitCollection?)`

Creates an image object using the named image asset that's compatible with the specified trait collection.

`init?(named: String, in: Bundle?, with: UIImage.Configuration?)`

Creates an image by using the named image asset that's compatible with the configuration you specify.

`convenience init?(named: String, in: Bundle?, variableValue: Double, configuration: UIImage.Configuration?)`

Creates an image by using the name, configuration, and variable value you specify.

`init?(named: String)`

Creates an image object from the specified named asset.

`convenience init(imageLiteralResourceName: String)`

Returns the image object for the specified resource.

`init?(systemName: String, withConfiguration: UIImage.Configuration?)`

Creates an image object that contains a system symbol image with the specified configuration.

`convenience init?(systemName: String, variableValue: Double, configuration: UIImage.Configuration?)`

Creates an image object that contains a system symbol image with the configuration and variable value you specify.

`init?(systemName: String, compatibleWith: UITraitCollection?)`

Creates an image object that contains a system symbol image appropriate for the specified traits.

`init?(systemName: String)`

Creates an image object that contains a system symbol image.

`convenience init(resource: ImageResource)`