Framework

Foundation

Access essential data types, collections, and operating-system services to define the base layer of functionality for your app.

iOS 2.0+ | iPadOS 2.0+ | Mac Catalyst 13.0+ | macOS 10.0+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

Overview

The Foundation framework provides a base layer of functionality for apps and frameworks, including data storage and persistence, text processing, date and time calculations, sorting and filtering, and networking. The classes, protocols, and data types defined by Foundation are used throughout the macOS, iOS, watchOS, and tvOS SDKs.

Topics

Fundamentals

- Numbers, Data, and Basic Values

  Work with primitive values and other fundamental types used throughout Cocoa.

- Strings and Text

  Create and process strings of Unicode characters, use regular expressions to find patterns, and perform natural language analysis of text.

- Collections

  Use arrays, dictionaries, sets, and specialized collections to store and iterate groups of objects or values.

- Dates and Times

Compare dates and times, and perform calendar and time zone calculations.

- Units and Measurement

  Label numeric quantities with physical dimensions to allow locale-aware formatting and conversion between related units.

- Data Formatting

  Convert numbers, dates, measurements, and other values to and from locale-aware string representations.

- Filters and Sorting

  Use predicates, expressions, and sort descriptors to examine elements in collections and other services.

App Support

- Task Management

  Manage your app's work and how it interacts with system services like Handoff and Shortcuts.

- Resources

  Access assets and other data bundled with your app.

- Notifications

  Design patterns for broadcasting information and for subscribing to broadcasts.

- App Extension Support

  Manage the interaction between an app extension and its hosting app.

- Errors and Exceptions

  Respond to problem situations in your interactions with APIs, and fine-tune your app for better debugging.

- Scripting Support

  Allow users to control your app with AppleScript and other automation technologies, or run scripts from within your app.

Files and Data Persistence

- File System

  Create, read, write, and examine files and folders in the file system.

- Archives and Serialization

  Convert objects and values to and from property list, JSON, and other flat binary representations.

- Preferences

  Persistently store domain-scoped pieces of information for configuring your app.

- Spotlight

  Search for files and other items on the local device, and index your app's content for searching.

- iCloud

  Manage files and key-value data that automatically synchronize among a user's iCloud devices.

- Optimizing Your App's Data for iCloud Backup

  Minimize the space and time that backups take to create by excluding purgeable and nonpurgeable data from backups.

Networking

- URL Loading System

  Interact with URLs and communicate with servers using standard Internet protocols.

- Bonjour

  Advertise services for easy discovery on local networks, or discover services advertised by others.

Low-Level Utilities

- XPC

  Manage secure interprocess communication.

- Object Runtime

  Get low-level support for basic Objective-C features, Cocoa design patterns, and Swift integration.

- Processes and Threads

  Manage your app's interaction with the host operating system and other processes, and implement low-level concurrency features.

- Streams, Sockets, and Ports

  Use low-level Unix features to manage input and output among files, processes, and the network.

Reference

- Foundation Enumerations
- Foundation Data Types

  This document describes the data types and constants found in the Foundation framework.

Articles

- allocWithZone:

  Returns an instance of the `NSMachPort` class.

- init

  Returns an initialized `NSOperation` object.

- initWithCoefficient:constant

  Initializes the unit converter with the coefficient and constant you specify.

- intersectSet:

  Removes from the receiving set each object that isn’t a member of another given set.

- isEqual:

  Returns a Boolean value that indicates whether the receiver and a given object have identical URL strings and base URLs.

- minusSet:

  Removes each object in another given set from the receiving set, if present.

- NSJavaBundleCleanup

  This function has been deprecated.

- NSJavaBundleSetup

  This function has been deprecated.

- NSJavaClassesForBundle

  Loads the Java classes located in the specified bundle.

NSJavaClassesFromPath

Loads the Java classes located at the specified path.

NSJavaDidCreateVirtualMachineNotification

Notification sent after the Java virtual machine is created.

NSJavaDidSetupVirtualMachineNotification

Notification sent after the Java virtual machine is set up.

NSJavaNeedsToLoadClasses

Returns a Boolean value that indicates whether a virtual machine is needed or if Java classes are provided.

NSJavaNeedsVirtualMachine

Returns a Boolean value that indicates whether a Java virtual machine is required.

NSJavaObjectNamedInPath

Creates an instance of the named class using the class loader previously specified at the given path.

NSJavaProvidesClasses

Returns a Boolean value that indicates whether Java classes are provided.

NSJavaSetup

Loads the Java virtual machine with specified parameters.

NSJavaSetupVirtualMachine

Sets up the Java virtual machine.

NSJavaWillCreateVirtualMachineNotification

Notification sent before the Java virtual machine is created.

NSJavaWillSetupVirtualMachineNotification

Notification sent before the Java virtual machine is set up.

NSLocaleSensitivePredicateOption

Indicates that strings to be compared using <, <=, =, =>, > should be handled in a locale-aware fashion.

NSMaxXEdge

The maximum X edge. This is equivalent to `CGRectEdge.maxXEdge`.

NSMaxYEdge

The maximum X edge. This is equivalent to `CGRectEdge.maxXEdge`.

NSMinXEdge

The minimum X edge. This is equivalent to `CGRectEdge.minXEdge`.

NSMinYEdge

The minimum Y edge. This is equivalent to `CGRectEdge.minYEdge`.

objCType

Returns a C string containing the Objective-C type of the data contained in the number object.

setURI:

Sets the URI of the receiver.

unionSet:

Adds each object in another given set to the receiving set, if not present.

Structures

`struct DiscontiguousAttributedSubstring`

A discontiguous portion of an attributed string.