

[Foundation](#) / [NSURLRequest](#)

Class

NSURLRequest

A URL load request that is independent of protocol or URL scheme.

iOS 2.0+ | iPadOS 2.0+ | Mac Catalyst 13.1+ | macOS 10.2+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
class NSURLRequest
```

Overview

Use this type in Swift when you need reference semantics or other Foundation-specific behavior.

[NSURLRequest](#) encapsulates two essential properties of a load request: the URL to load and the policies used to load it. In addition, for HTTP and HTTPS requests, [URLRequest](#) includes the HTTP method (GET, POST, and so on) and the HTTP headers. Finally, custom protocols can support custom properties as explained in [Custom protocol properties](#).

[NSURLRequest](#) only represents information about the request. Use other classes, such as [URLSession](#), to send the request to a server. See [Fetching website data into memory](#) and [Uploading data to a website](#) for an introduction to these techniques.

The mutable subclass of [NSURLRequest](#) is [NSMutableURLRequest](#).

Important

The Swift overlay to the Foundation framework provides the [URLRequest](#) structure, which bridges to the [NSURLRequest](#) class and its mutable subclass, [NSMutableURLRequest](#). For more information about value types, see [Working with Cocoa Frameworks](#) in [Using Swift with Cocoa and Objective-C \(Swift 4.1\)](#).

Reserved HTTP headers

The URL Loading System handles various aspects of the HTTP protocol for you (HTTP 1.1 persistent connections, proxies, authentication, and so on). As part of this support, the URL Loading System takes responsibility for certain HTTP headers:

- Content-Length
- Authorization
- Connection
- Host
- Proxy-Authenticate
- Proxy-Authorization
- WWW-Authenticate

If you set a value for one of these reserved headers, the system may ignore the value you set, or overwrite it with its own value, or simply not send it. Moreover, the exact behavior may change over time. To avoid confusing problems like this, do not set these headers directly.

The URL Loading System sets the Content-Length header based on whether the request body has a known length:

- If so, it uses the identity transfer encoding and sets the Content-Length header to that known length. You see this when you set the request body to a data object.
- If not, it uses the chunked transfer encoding and omits the Content-Length header. You see this when you set the request body to a stream.

Custom protocol properties

If you implement a custom URL protocol by subclassing `URLProtocol`, and it needs protocol-specific properties, extend `NSURLRequest` with accessor methods for those custom properties. In your accessor methods, call `property(forKey:in:)` and `setProperty(_:forKey:in:)` to associate property values with the request.

Topics

Creating requests

```
convenience init(url: URL)
```

Creates a URL request for a specified URL.

```
init(url: URL, cachePolicy: NSURLRequest.CachePolicy, timeoutInterval: TimeInterval)
```

Creates a URL request with the specified URL, cache policy, and timeout values.

Working with a cache policy

```
var cachePolicy: NSURLRequest.CachePolicy
```

The request's cache policy.

```
enum CachePolicy
```

The constants used to specify interaction with the cached responses.

Accessing request components

```
var httpMethod: String?
```

The HTTP request method.

```
var url: URL?
```

The URL being requested.

```
var httpBody: Data?
```

The request body.

```
var httpBodyStream: InputStream?
```

The request body as an input stream.

```
var mainDocumentURL: URL?
```

The main document URL associated with the request.

Getting header fields

```
var allHTTPHeaderFields: [String : String]?
```

A dictionary containing all of the HTTP header fields for a request.

```
func value(forHTTPHeaderField: String) -> String?
```

Returns the value of the specified HTTP header field.

Controlling request behavior

```
var timeoutInterval: TimeInterval
```

The request's timeout interval, in seconds.

```
var httpShouldHandleCookies: Bool
```

A Boolean value that indicates whether the default cookie handling will be used for this request.

```
var httpShouldUsePipelining: Bool
```

A Boolean value that indicates whether the request should continue transmitting data before receiving a response from an earlier transmission.

Deprecated

```
var allowsCellularAccess: Bool
```

A Boolean value that indicates whether the request is allowed to use the cellular radio (if present).

Supporting limited modes

```
var allowsConstrainedNetworkAccess: Bool
```

A Boolean value that indicates whether connections may use the network when the user has specified Low Data Mode.

```
var allowsExpensiveNetworkAccess: Bool
```

A Boolean value that indicates whether connections may use a network interface that the system considers expensive.

Accessing the service type

```
var networkServiceType: NSURLRequest.NetworkServiceType
```

The network service type of the request.

```
enum NetworkServiceType
```

Constants that specify how a request uses network resources.

Supporting secure coding

```
class var supportsSecureCoding: Bool
```

A Boolean value indicating whether the NSURLRequest implements the NSSecureCoding protocol.

Indicating the source of the request

`var attribution: NSURLRequest.Attribution`

The entity that initiates the network request.

`enum Attribution`

The entities that can make a network request.

Instance Properties

`var allowsPersistentDNS: Bool`

`var allowsUltraConstrainedNetworkAccess: Bool`

`var assumesHTTP3Capable: Bool`

`var cookiePartitionIdentifier: String?`

`var requiresDNSSECValidation: Bool`

Relationships

Inherits From

`NSObject`

Inherited By

`NSMutableURLRequest`

Conforms To

`CVarArg`

`Copyable`

`CustomDebugStringConvertible`

`CustomStringConvertible`

`Equatable`

`Hashable`

`NSCoding`

`NSCopying`

`NSMutableCopying`

`NSObjectProtocol`

See Also

Requests and responses

`struct URLRequest`

A URL load request that is independent of protocol or URL scheme.

`class NSMutableURLRequest`

A mutable URL load request that is independent of protocol or URL scheme.

`class URLResponse`

The metadata associated with the response to a URL load request, independent of protocol and URL scheme.

`class HTTPURLResponse`

The metadata associated with the response to an HTTP protocol URL load request.