

[Foundation](#) / URL Loading System

API Collection

URL Loading System

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

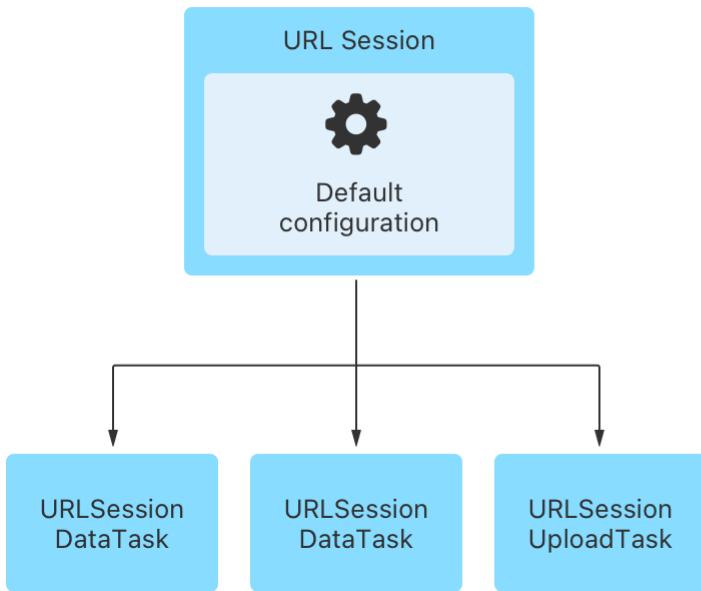
Overview

The URL Loading System provides access to resources identified by URLs, using standard protocols like `https` or custom protocols you create. Loading is performed asynchronously, so your app can remain responsive and handle incoming data or errors as they arrive.

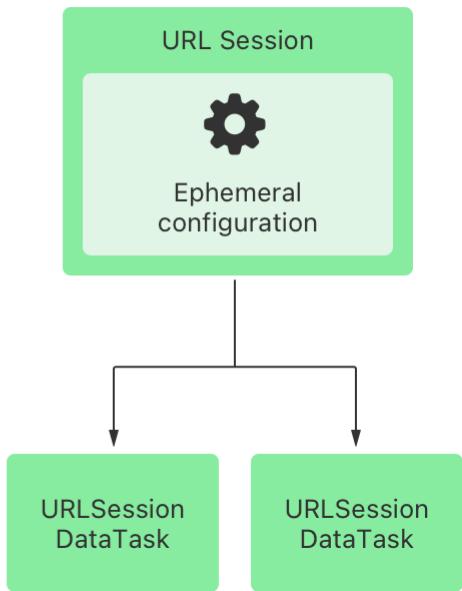
You use a [`NSURLSession`](#) instance to create one or more [`NSURLSessionTask`](#) instances, which can fetch and return data to your app, download files, or upload data and files to remote locations. To configure a session, you use a [`NSURLSessionConfiguration`](#) object, which controls behavior like how to use caches and cookies, or whether to allow connections on a cellular network.

You can use one session repeatedly to create tasks. For example, a web browser might have separate sessions for regular and private browsing use, where the private session doesn't cache its data. [Figure 1](#) shows how two sessions with these configurations can then create multiple tasks.

Default browsing



Private browsing



Each session is associated with a delegate to receive periodic updates (or errors). The default delegate calls a completion handler block that you provide; if you choose to provide your own custom delegate, this block is not called.

You can configure a session to run in the background, so that while the app is suspended, the system can download data on its behalf and wake up the app to deliver the results.

Topics

Essentials

Configure and create sessions, then use them to create tasks that interact with URLs.

Fetching website data into memory

Receive data directly into memory by creating a data task from a URL session.

Analyzing HTTP traffic with Instruments

Measure HTTP-based network performance and usage of your apps.

`class URLSession`

An object that coordinates a group of related, network data transfer tasks.

`class URLSessionTask`

A task, like downloading a specific resource, performed in a URL session.

Requests and responses

```
struct URLRequest
```

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

```
class NSURLRequest
```

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.

Uploading

{ } Building a resumable upload server with SwiftNIO

Support HTTP resumable upload protocol in SwiftNIO by translating resumable uploads to regular uploads.

📄 Uploading data to a website

Post data from your app to servers.

📄 Uploading streams of data

Send a stream of data to a server.

📄 Pausing and resuming uploads

Pause and resume an upload without starting over, even when the connection is interrupted.

Downloading

📄 Downloading files from websites

Download files directly to the filesystem.

📄 Pausing and resuming downloads

Allow the user to resume a download without starting over.

📄 Downloading files in the background

Create tasks that download files while your app is inactive.

Cache behavior

Accessing cached data

Control how URL requests make use of previously cached data.

`class CachedURLResponse`

A cached response to a URL request.

`class URLCache`

An object that maps URL requests to cached response objects.

Authentication and credentials

Handling an authentication challenge

Respond appropriately when a server demands authentication for a URL request.

`class URLAuthenticationChallenge`

A challenge from a server requiring authentication from the client.

`class URLCredential`

An authentication credential consisting of information specific to the type of credential and the type of persistent storage to use, if any.

`class URLCredentialStorage`

The manager of a shared credentials cache.

`class URLProtectionSpace`

A server or an area on a server, commonly referred to as a realm, that requires authentication.

Network activity attribution

Inspecting app activity data

Verify that your app accesses only the user data and network resources that you expect it to access.

Indicating the source of network activity

Control whether the App Privacy Report attributes network traffic to the app or to the user.

Cookies

```
class HTTPCookie
```

A representation of an HTTP cookie.

```
class HTTPCookieStorage
```

A container that manages the storage of cookies.

Errors

```
struct URLError
```

Error codes returned by URL loading APIs.

≡ URL Loading System error info keys

Recognize these keys from the user info dictionary of error objects produced by URL Loading APIs.

Legacy

≡ Legacy URL Loading Systems

Migrate your code away from using these legacy objects.

See Also

Networking

≡ Bonjour

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