Discover   Design   Develop   Distribute   Support   Account

Documentation ···
                      URLSession
                      URLSessionDownloadTask
              Language: Swift ⌄
              API Changes: None

URLSessionDownloadTask

A URL session task that stores downloaded data to file.

Declaration

```
class URLSessionDownloadTask : URLSessionTask
```

Overview

An `URLSessionDownloadTask` is a concrete subclass of `URLSessionTask`. Most of the methods associated with this class are documented in `URLSessionTask`.

Download tasks directly write the server's response data to a temporary file, providing your app with progress updates as data arrives from the server. When you use download tasks in background sessions, these downloads continue even when your app is suspended or is otherwise not running.

You can pause (cancel) download tasks and resume them later (assuming the server supports doing so). You can also resume downloads that failed because of network connectivity problems.

Download Delegate Behavior

When you use a download task, your delegate receives several callbacks unique to download scenarios.

- During download, the session periodically calls the delegate's `urlSession(_:downloadTask:didWriteData:totalBytesWritten:totalBytesExpectedToWrite:)` method with status information.

- Upon successful completion, the session calls the delegate's `urlSession(_:downloadTask:didFinishDownloadingTo:)` method or completion handler. In that method, you must either open the file for reading or move it to a permanent location in your app's sandbox container directory.

- Upon unsuccessful completion, the session calls the delegate's `urlSession(_:task:didCompleteWithError:)` method or completion handler. Unlike `URLSessionData`

SDKs

iOS 7.0+

macOS 10.9+

tvOS 9.0+

watchOS 2.0+

Framework

Foundation

On This Page

Declaration ⌄
Overview ⌄
Topics ⌄
Relationships ⌄
See Also ⌄

`Task` or `URLSessionUploadTask`, a download task reports server-side errors reported through HTTP status codes into corresponding `NSError` objects. These errors are shown in Table 1.

Table 1 NSError Codes for Server-Side Download Errors

HTTPS Status	NSError code
`401 Unauthorized`	`NSURLErrorUserAuthentication Required`
`403 Unauthorized`	`NSURLErrorNoPermissionsToReadFile`
`407 Proxy Authentication Required`	`NSURLErrorUserAuthentication Required`
(other)	`NSURLErrorFileDoesNotExist`

Topics

Canceling a Download

`func cancel(byProducingResumeData: (Data?) -> Void)`

Cancels a download and calls a callback with resume data for later use.

Relationships

Inherits From

`URLSessionTask`

Conforms To

`CVarArg`, `Equatable`, `Hashable`

See Also

Adding Download Tasks to a Session

```
func downloadTask(with: URL) -> URLSessionDownloadTask
```
Creates a download task that retrieves the contents of the specified URL and saves the results to a file.

```
func downloadTask(with: URL, completionHandler: (URL?, URLResponse?, Error?) -> Void) -> URLSessionDownloadTask
```
Creates a download task that retrieves the contents of the specified URL, saves the results to a file, and calls a handler upon completion.

```
func downloadTask(with: URLRequest) -> URLSessionDownloadTask
```
Creates a download task that retrieves the contents of a URL based on the specified URL request object and saves the results to a file.

```
func downloadTask(with: URLRequest, completionHandler: (URL?, URLResponse?, Error?) -> Void) -> URLSessionDownloadTask
```
Creates a download task that retrieves the contents of a URL based on the specified URL request object, saves the results to a file, and calls a handler upon completion.

```
func downloadTask(withResumeData: Data) -> URLSessionDownloadTask
```
Creates a download task to resume a previously canceled or failed download.

```
func downloadTask(withResumeData: Data, completionHandler: (URL?, URLResponse?, Error?) -> Void) -> URLSessionDownloadTask
```
Creates a download task to resume a previously canceled or failed download and calls a handler upon completion.

```
protocol URLSessionDownloadDelegate
```
A protocol defining methods that URL session instances call on their delegates to handle task-level events specific to download tasks.