Discover Design Develop Distribute Support Account

Documentation ···
URLSession
URLSessionTaskDelegate
Language: Swift
API Changes: None

URLSessionTaskDelegate

A protocol defining methods that URL session instances call on their delegates to handle task-level events.

SDKs

iOS 7.0+

macOS 10.9+

tvOS 9.0+

watchOS 2.0+

Framework

Foundation

On This Page

Declaration
Overview
Topics
Relationships
See Also

Declaration

```
protocol URLSessionTaskDelegate
```

Overview

If you’re using download tasks, also implement the methods in the `URLSessionDownloadDelegate` protocol.

If you’re using data or upload tasks, also implement the methods in the `URLSessionDataDelegate` protocol.

> **Note**
>
> Your `URLSession` object doesn’t need to have a delegate. If no delegate is assigned, a system-provided delegate is used, and you must provide a completion callback to obtain the data.

Topics

Handling Task Life Cycle Changes

```
func urlSession(URLSession, task: URLSessionTask, didCompleteWithError: Error?)
```

Tells the delegate that the task finished transferring data.

Handling Redirects

```
func urlSession(URLSession, task: URLSessionTask, willPerformHTTPRedirection: HTTPURLResponse, newRequest: URLRequest, completionHandler: (URLRequest?) -> Void)
```

Tells the delegate that the remote server requested an HTTP redirect.

Working with Upload Tasks

```
func urlSession(URLSession, task: URLSessionTask, didSendBodyData: Int64, totalBytesSent: Int64, totalBytesExpectedToSend: Int64)
```

Periodically informs the delegate of the progress of sending body content to the server.

```
func urlSession(URLSession, task: URLSessionTask, needNewBodyStream: (InputStream?) -> Void)
```

Tells the delegate when a task requires a new request body stream to send to the remote server.

Handling Authentication Challenges

```
func urlSession(URLSession, task: URLSessionTask, didReceive: URLAuthenticationChallenge, completionHandler: (URLSession.AuthChallengeDisposition, URLCredential?) -> Void)
```

Requests credentials from the delegate in response to an authentication request from the remote server.

```
enum URLSession.AuthChallengeDisposition
```

Constants passed by session or task delegates to the provided continuation block in response to an authentication challenge.

Handling Delayed and Waiting Tasks

```
func urlSession(URLSession, task: URLSessionTask, willBeginDelayedRequest: URLRequest, completionHandler: (URLSession.DelayedRequestDisposition, URLRequest?) -> Void)
```

Tells the delegate that a delayed URL session task will now begin loading.

```
enum URLSession.DelayedRequestDisposition
```

The action to take on a delayed URL session task.

```
func urlSession(URLSession, taskIsWaitingForConnectivity: URLSessionTask)
```

Tells the delegate that the task is waiting until suitable connectivity is available before beginning the network load.

Collecting Task Metrics

```
func urlSession(URLSession, task: URLSessionTask, didFinish
Collecting: URLSessionTaskMetrics)
```

Tells the delegate that the session finished collecting metrics for the task.

```
class URLSessionTaskMetrics
```

An object encapsulating the metrics for a session task.

Relationships

Inherits From

```
URLSessionDelegate
```

Inherited By

```
AVAssetDownloadDelegate
URLSessionDataDelegate
URLSessionDownloadDelegate
URLSessionStreamDelegate
```

See Also

Working with a Delegate

```
var delegate: URLSessionDelegate?
```

The delegate assigned when this object was created.

```
protocol URLSessionDelegate
```

A protocol defining methods that URL session instances call on their delegates to handle session-level events, like session life cycle changes.

```
var delegateQueue: OperationQueue
```

The operation queue provided when this object was created.