

urlSession(_ session: URLSession, task: URLSessionTask, willBeginDelayedRequest request: URLRequest, completionHandler: @escaping (URLSession.DelayedRequestDisposition, URLRequest?) -> Void)

urlSession(_ session: URLSession, taskIsWaitingForConnectivity task: URLSessionTask)

urlSession(_ session: URLSession, task: URLSessionTask, willPerformHTTPRedirection response: HTTPURLResponse, newRequest request: URLRequest, completionHandler: @escaping (URLRequest?) -> Void)

urlSession(_ session: URLSession, task: URLSessionTask, didReceive challenge: URLAuthenticationChallenge, completionHandler: @escaping (URLSession.AuthChallengeDisposition, URLCredential?) -> Void)

urlSession(_ session: URLSession, task: URLSessionTask, needNewBodyStream completionHandler: @escaping (InputStream?) -> Void)

urlSession(_ session: URLSession, task: URLSessionTask, didSendBodyData bytesSent: Int64, totalBytesSent: Int64, totalBytesExpectedToSend: Int64)

urlSession(_ session: URLSession, task: URLSessionTask, didFinishCollecting metrics: URLSessionTaskMetrics)

urlSession(_ session: URLSession, task: URLSessionTask, didCompleteWithError error: Error?)

代理方法

urlSession(_ session: URLSession, didBecomeInvalidWithError error: Error?)

urlSession(_ session: URLSession, didReceive challenge: URLAuthenticationChallenge, completionHandler: @escaping (URLSession.AuthChallengeDisposition, URLCredential?) -> Void)

urlSessionDidFinishEvents(forBackgroundURLSession session: URLSession)

代理方法

urlSession(_ session: URLSession, dataTask: URLSessionDataTask, didReceive response: URLResponse, completionHandler: @escaping (URLSession.ResponseDisposition) -> Void)

urlSession(_ session: URLSession, dataTask: URLSessionDataTask, didBecome downloadTask: URLSessionDownloadTask)

urlSession(_ session: URLSession, dataTask: URLSessionDataTask, didBecome streamTask: URLSessionStreamTask)

urlSession(_ session: URLSession, dataTask: URLSessionDataTask, didReceive data: Data)

urlSession(_ session: URLSession, dataTask: URLSessionDataTask, willCacheResponse proposedResponse: CachedURLResponse, completionHandler: @escaping (CachedURLResponse?) -> Void)

代理方法

urlSession(_ session: URLSession, downloadTask: URLSessionDownloadTask, didFinishDownloadingTo location: URL)

urlSession(_ session: URLSession, downloadTask: URLSessionDownloadTask, didWriteData bytesWritten: Int64, totalBytesWritten: Int64, totalBytesExpectedToWrite: Int64)

urlSession(_ session: URLSession, downloadTask: URLSessionDownloadTask, didResumeAtOffset fileOffset: Int64, expectedTotalBytes: Int64)

代理方法

urlSession(_ session: URLSession, readClosedFor streamTask: URLSessionStreamTask)

urlSession(_ session: URLSession, writeClosedFor streamTask: URLSessionStreamTask)

urlSession(_ session: URLSession, betterRouteDiscoveredFor streamTask: URLSessionStreamTask)

urlSession(_ session: URLSession, streamTask: URLSessionStreamTask, didBecome inputStream: InputStream, outputStream: OutputStream)

代理方法

urlSession(_ session: URLSession, websocketTask: URLSessionWebSocketTask, didOpenWithProtocol protocol: String?)

urlSession(_ session: URLSession, websocketTask: URLSessionWebSocketTask, didCloseWith closeCode: URLSessionWebSocketTask.CloseCode, reason: Data?)

代理方法

NSObjectProtocol

URLSessionDelegate

URLSessionTaskDelegate

URLSessionDataDelegate

URLSessionDownloadDelegate

URLSessionStreamDelegate

URLSessionWebSocketDelegate