NSURLProtocolClient Protocol Reference

Cocoa > Networking

2006-05-23

Apple Inc.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

NSURLProtocolClient Protocol Reference

Conforms to	NSObject
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.2 with Safari 1.0 installed. Available in Mac OS X v10.2.7 and later.
Companion guide	URL Loading System
Declared in	NSURLProtocol.h

Overview

The `NSURLProtocolClient` protocol provides the interface used by `NSURLProtocol` subclasses to communicate with the URL loading system. An application should never have the need to implement this protocol.

Tasks

Protocol Methods

`– URLProtocol:cachedResponseIsValid:` (page 6)
Sent to indicate to the URL loading system that a cached response is valid.

`– URLProtocol:didCancelAuthenticationChallenge:` (page 6)
Sent to indicate to the URL loading system that an authentication challenge has been canceled.

`– URLProtocol:didFailWithError:` (page 7)
Sent when the load request fails due to an error.

`– URLProtocol:didLoadData:` (page 7)
An NSURLProtocol subclass instance, *`protocol`*, sends this message to `[protocol client]` as it loads *`data`*.

`– URLProtocol:didReceiveAuthenticationChallenge:` (page 7)
Sent to indicate to the URL loading system that an authentication challenge has been received.

`– URLProtocol:didReceiveResponse:cacheStoragePolicy:` (page 8)
Sent to indicate to the URL loading system that the protocol implementation has created a response object for the request.

`- URLProtocol:wasRedirectedToRequest:redirectResponse:` (page 8)
: Sent to indicate to the URL loading system that the protocol implementation has been redirected.

`- URLProtocolDidFinishLoading:` (page 9)
: Sent to indicate to the URL loading system that the protocol implementation has finished loading.

Instance Methods

URLProtocol:cachedResponseIsValid:

Sent to indicate to the URL loading system that a cached response is valid.

```
- (void)URLProtocol:(NSURLProtocol *)protocol
    cachedResponseIsValid:(NSCachedURLResponse *)cachedResponse
```

Parameters

protocol

The URL protocol object sending the message.

cachedResponse

The cached response whose validity is being communicated.

Availability

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

Declared In

`NSURLProtocol.h`

URLProtocol:didCancelAuthenticationChallenge:

Sent to indicate to the URL loading system that an authentication challenge has been canceled.

```
- (void)URLProtocol:(NSURLProtocol *)protocol
    didCancelAuthenticationChallenge:(NSURLAuthenticationChallenge *)challenge
```

Parameters

protocol

The URL protocol object sending the message.

challenge

The authentication challenge that was canceled.

Availability

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

Declared In

`NSURLProtocol.h`

URLProtocol:didFailWithError:

Sent when the load request fails due to an error.

```
- (void)URLProtocol:(NSURLProtocol *)protocol didFailWithError:(NSError *)error
```

Parameters

protocol

The URL protocol object sending the message.

error

The error that caused the failure of the load request.

Availability

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

Declared In

`NSURLProtocol.h`

URLProtocol:didLoadData:

An NSURLProtocol subclass instance, *protocol*, sends this message to `[protocol client]` as it loads *data*.

```
- (void)URLProtocol:(NSURLProtocol *)protocol didLoadData:(NSData *)data
```

Discussion

The data object must contain only new data loaded since the previous invocation of this method.

Availability

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

Declared In

`NSURLProtocol.h`

URLProtocol:didReceiveAuthenticationChallenge:

Sent to indicate to the URL loading system that an authentication challenge has been received.

```
- (void)URLProtocol:(NSURLProtocol *)protocol
    didReceiveAuthenticationChallenge:(NSURLAuthenticationChallenge *)challenge
```

Parameters

protocol

The URL protocol object sending the message.

challenge

The authentication challenge that has been received.

Discussion

The protocol client guarantees that it will answer the request on the same thread that called this method. The client may add a default credential to the challenge it issues to the connection delegate, if *protocol* did not provide one.

Availability
Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

Declared In
`NSURLProtocol.h`

URLProtocol:didReceiveResponse:cacheStoragePolicy:

Sent to indicate to the URL loading system that the protocol implementation has created a response object for the request.

```
- (void)URLProtocol:(NSURLProtocol *)protocol didReceiveResponse:(NSURLResponse
    *)response cacheStoragePolicy:(NSURLCacheStoragePolicy)policy
```

Parameters

protocol
: The URL protocol object sending the message.

response
: The newly available response object.

policy
: The cache storage policy for the response.

Discussion
The implementation should provide the NSURLCacheStoragePolicy that should be used if the response is to be stored in a cache as the *policy* value.

Availability
Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

Declared In
`NSURLProtocol.h`

URLProtocol:wasRedirectedToRequest:redirectResponse:

Sent to indicate to the URL loading system that the protocol implementation has been redirected.

```
- (void)URLProtocol:(NSURLProtocol *)protocol wasRedirectedToRequest:(NSURLRequest
    *)request redirectResponse:(NSURLResponse *)redirectResponse
```

Parameters

protocol
: The URL protocol object sending the message.

request
: The new request that the original request was redirected to.

redirectResponse
: The response from the original request that caused the redirect.

Availability
Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

Declared In
`NSURLProtocol.h`

URLProtocolDidFinishLoading:

Sent to indicate to the URL loading system that the protocol implementation has finished loading.

```
- (void)URLProtocolDidFinishLoading:(NSURLProtocol *)protocol
```

Parameters

protocol
The URL protocol object sending the message.

Availability
Available in Mac OS X v10.2 with Safari 1.0 installed.
Available in Mac OS X v10.2.7 and later.

Declared In
`NSURLProtocol.h`

Document Revision History

This table describes the changes to *NSURLProtocolClient Protocol Reference*.

Date	Notes
2006-05-23	First publication of this content as a separate document.

Index