WebResourceLoadDelegate Protocol Reference

Cocoa > User Experience

2008-06-19

Apple Inc.
© 2008 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

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

Apple, the Apple logo, Cocoa, Mac, Mac OS, Objective-C, and Safari are trademarks of Apple Inc., registered in the United States and other countries.

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

WebResourceLoadDelegate Protocol Reference

(informal protocol)

Framework	/System/Library/Frameworks/WebKit.framework
Availability	Available in Mac OS X v10.2 with Safari 1.0 and later. Available in Mac OS X v10.2.7 and later.
Companion guide	WebKit Objective-C Programming Guide
Declared in	WebResourceLoadDelegate.h

Overview

WebView resource load delegates implement this informal protocol to be notified on the progress of loading individual resources. Note that there can be hundreds of resources, such as images and other media, per page. So, if you just want to get page loading status see the `WebFrameLoadDelegate`protocol.

There's a separate client request and server response made for each resource on a page. By implementing the `webView:identifierForInitialRequest:fromDataSource:` (page 6) method, resource load delegates provide a tracking object used to identify individual resources in subsequent calls to delegate methods. Delegates are then notified when resource loading starts, when data is incrementally received, when any load errors occur, and when the load is complete. Delegates may also change a request before it is sent. In some cases, depending on the page content and server redirects, methods defined in this protocol may be invoked multiple times (see individual method descriptions for more details). All the methods in this protocol are optional.

Tasks

Setting Identifiers

- `webView:identifierForInitialRequest:fromDataSource:` (page 6)
  Returns an identifier object used to track the progress of loading a single resource.

Loading Content

- `webView:resource:willSendRequest:redirectResponse:fromDataSource:` (page 10)
  Invoked before a request is initiated for a resource and returns a possibly modified request.
- `webView:resource:didFinishLoadingFromDataSource:` (page 8)
  Invoked when all of the data for a given resource is loaded.

- `webView:resource:didReceiveResponse:fromDataSource:` (page 9)
  Invoked after a resource has been loaded.
- `webView:resource:didReceiveContentLength:fromDataSource:` (page 9)
  Invoked when some of the data for a given resource has arrived.
- `webView:resource:didFailLoadingWithError:fromDataSource:` (page 7)
  Invoked when a resource failed to load.
- `webView:plugInFailedWithError:dataSource:` (page 6)
  Invoked when a plug-in fails to load.

Authenticating Resources

- `webView:resource:didReceiveAuthenticationChallenge:fromDataSource:` (page 8)
  Invoked when an authentication challenge has been received for a resource.
- `webView:resource:didCancelAuthenticationChallenge:fromDataSource:` (page 7)
  Invoked when an authentication challenge for a resource was cancelled.

Instance Methods

webView:identifierForInitialRequest:fromDataSource:

Returns an identifier object used to track the progress of loading a single resource.

```
- (id)webView:(WebView *)sender identifierForInitialRequest:(NSURLRequest *)request
     fromDataSource:(WebDataSource *)dataSource
```

Discussion
The identifier returned by this delegate method will be retained by *sender* and passed as an argument to all other delegate messages pertaining to this resource. The *request* argument is the request that initiated this load for *dataSource*. Delegates might implement this method to begin tracking the progress of loading an individual resource. Note that this method is invoked once per load where as `webView:resource:willSendRequest:redirectResponse:fromDataSource:` (page 10) may be invoked multiple times.

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

Available in Mac OS X v10.2.7 and later.

Declared In
`WebResourceLoadDelegate.h`

webView:plugInFailedWithError:dataSource:

Invoked when a plug-in fails to load.

```
- (void)webView:(WebView *)sender plugInFailedWithError:(NSError *)error
    dataSource:(WebDataSource *)dataSource
```

Discussion

For example, this method is invoked if a plug-in is not found, fails to load, or is not available for some reason. The *error* argument is the error that occurred during the process of loading that resource. Delegates might implement this method to display or log a detailed error message. If you do not implement this method, no action is taken.

The *userInfo* dictionary of *error* may contain additional information about the failure. If the `userInfo` dictionary is not `nil`, it may contain some or all of these key-value pairs. The value of the `NSErrorFailingURLKey` key will be a URL string of the `SRC` attribute. The value of the `WebKitErrorPlugInNameKey` key is will be a string containing the plug-in's name. The value for the `WebKitErrorPlugInPageURLStringKey` key will be a URL string of the `PLUGINSPAGE` attribute. The value of the `WebKitErrorMIMETypeKey` key will be a string of the `TYPE` attribute.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

`- webView:resource:didFailLoadingWithError:fromDataSource:` (page 7)

Declared In

`WebResourceLoadDelegate.h`

webView:resource:didCancelAuthenticationChallenge:fromDataSource:

Invoked when an authentication challenge for a resource was cancelled.

```
- (void)webView:(WebView *)sender resource:(id)identifier
    didCancelAuthenticationChallenge:(NSURLAuthenticationChallenge *)challenge
    fromDataSource:(WebDataSource *)dataSource
```

Discussion

The *identifier* argument is used to track the resource being loaded by *dataSource*, and *challenge* is the authentication challenge that was cancelled.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

`- webView:resource:didReceiveAuthenticationChallenge:fromDataSource:` (page 8)

Declared In

`WebResourceLoadDelegate.h`

webView:resource:didFailLoadingWithError:fromDataSource:

Invoked when a resource failed to load.

```
- (void)webView:(WebView *)sender resource:(id)identifier
    didFailLoadingWithError:(NSError *)error fromDataSource:(WebDataSource
    *)dataSource
```

2008-06-19 | © 2008 Apple Inc. All Rights Reserved.

Discussion

The *identifier* argument is used to track the resource being loaded by *dataSource*, and *error* is the error that occurred loading that resource. Delegates might implement this method to display or log a detailed error message.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

- `webView:plugInFailedWithError:dataSource:` (page 6)

Declared In

`WebResourceLoadDelegate.h`

webView:resource:didFinishLoadingFromDataSource:

Invoked when all of the data for a given resource is loaded.

```
- (void)webView:(WebView *)sender resource:(id)identifier
    didFinishLoadingFromDataSource:(WebDataSource *)dataSource
```

Discussion

The *identifier* argument is used to track the resource being loaded by *dataSource*. Delegates might implement this method to update the load status of an individual resource.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

- `webView:resource:willSendRequest:redirectResponse:fromDataSource:` (page 10)

Declared In

`WebResourceLoadDelegate.h`

webView:resource:didReceiveAuthenticationChallenge:fromDataSource:

Invoked when an authentication challenge has been received for a resource.

```
- (void)webView:(WebView *)sender resource:(id)identifier
    didReceiveAuthenticationChallenge:(NSURLAuthenticationChallenge *)challenge
    fromDataSource:(WebDataSource *)dataSource
```

Discussion

The *identifier* argument is used to track the resource being loaded by *dataSource*, and *challenge* is the authentication challenge that was received.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

– `webView:resource:didCancelAuthenticationChallenge:fromDataSource:` (page 7)

Declared In

`WebResourceLoadDelegate.h`

webView:resource:didReceiveContentLength:fromDataSource:

Invoked when some of the data for a given resource has arrived.

```
- (void)webView:(WebView *)sender resource:(id)identifier
    didReceiveContentLength:(NSUInteger)length fromDataSource:(WebDataSource
    *)dataSource
```

Discussion

The *identifier* argument is used to track the resource being loaded by *dataSource*, and *length* is the amount of incremental data received for this resource—the amount of data loaded since the last time this method was invoked for this resource, not the total amount received for this resource. Delegates might implement this method to update the load status of an individual resource.

The *length* parameter was changed from an `unsigned int` to an `NSUInteger` in Mac OS X v10.5.

Availability

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

Available in Mac OS X v10.2.7 and later.

Declared In

`WebResourceLoadDelegate.h`

webView:resource:didReceiveResponse:fromDataSource:

Invoked after a resource has been loaded.

```
- (void)webView:(WebView *)sender resource:(id)identifier
    didReceiveResponse:(NSURLResponse *)response fromDataSource:(WebDataSource
    *)dataSource
```

Discussion

The *identifier* argument is used to track the resource being loaded by *dataSource*, and *response* is the reply that was received.

In some rare cases, multiple responses may be received for a single resource. This happens in the case of multipart/x-mixed-replace, also known as a "server push." In this case, delegates should assume that the progress of loading this resource restarts, and the expected content length may change.

Availability

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

Available in Mac OS X v10.2.7 and later.

Declared In

`WebResourceLoadDelegate.h`

webView:resource:willSendRequest:redirectResponse:fromDataSource:

Invoked before a request is initiated for a resource and returns a possibly modified request.

```
- (NSURLRequest *)webView:(WebView *)sender resource:(id)identifier
    willSendRequest:(NSURLRequest *)request redirectResponse:(NSURLResponse
    *)redirectResponse fromDataSource:(WebDataSource *)dataSource
```

Discussion

The `identifier` argument is used to track the resource being loaded by `dataSource`. The `request` argument is the request that will be sent, and `redirectResponse` is the redirect server response. The `redirectResponse` argument is `nil` if there is no redirect in progress. Delegates might implement this method to modify resource requests before they are sent. Note that this method might be invoked multiple times per load (as a result of a server redirect) where as `webView:identifierForInitialRequest:fromDataSource:` (page 6) will be invoked once.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

`- webView:resource:didFinishLoadingFromDataSource:` (page 8)

Declared In

`WebResourceLoadDelegate.h`

Document Revision History

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

Date	Notes
2008-06-19	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Index