
WebDataSource Class Reference

[Cocoa](#) > [User Experience](#)



2008-10-15



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, Carbon, 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

WebDataSource Class Reference 5

Overview	5
Tasks	5
Initializing an Instance	5
Querying Page Data and State	6
Getting the Request and Response	6
Getting the Web Frame	6
Getting an Unreachable URL	6
Getting a Web Archive	6
Accessing Subresources	6
Instance Methods	7
addSubresource:	7
data	7
initialRequest	8
initWithRequest:	8
isLoading	8
mainResource	9
pageTitle	9
representation	9
request	10
response	10
subresourceForURL:	11
subresources	11
textEncodingName	11
unreachableURL	12
webArchive	12
webFrame	12

Document Revision History 15

Index 17

WebDataSource Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
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	WebDataSource.h
Related sample code	NewsReader SpecialPictureProtocol

Overview

WebDataSource encapsulates the web content to be displayed in a web frame view. A WebDataSource object has a representation object, conforming to the WebDocumentRepresentation protocol, that holds the data in an appropriate format depending on the MIME type. You can extend WebKit to support new MIME types by implementing your own view and representation classes, and specifying the mapping between them using the `WebView` `registerViewClass:representationClass:forMIMEType:` class method.

WebDataSource objects have an associated initial request, possibly modified request, and response object. Since the data source may be in the process of being loaded, you should check the state of a data source using the `isLoading` (page 8) method before accessing its data. Use the `data` (page 7) method to get the raw data. Use the `representation` (page 9) method to get the actual representation object and query it for more details.

Tasks

Initializing an Instance

- `initWithRequest:` (page 8)
initializes a data source with a URL request.

Querying Page Data and State

- [data](#) (page 7)
Returns the raw data that represents the receiver's content.
- [isLoading](#) (page 8)
Returns YES if the receiver is in the process of loading its content, NO otherwise.
- [pageTitle](#) (page 9)
Returns the title of the receiver's page.
- [representation](#) (page 9)
Returns the receiver's representation depending on its MIME type.
- [textEncodingName](#) (page 11)
Returns either the text encoding for the receiver's WebView, if set, or the text encoding of the response.

Getting the Request and Response

- [initialRequest](#) (page 8)
Returns a reference to the original request that was used to load the web content.
- [request](#) (page 10)
Returns the request that was used to create the receiver.
- [response](#) (page 10)
Returns the associated WebResourceResponse for this data source.

Getting the Web Frame

- [webFrame](#) (page 12)
Returns the web frame that represents this data source.

Getting an Unreachable URL

- [unreachableURL](#) (page 12)
Returns the receiver's unreachable URL if it exists, nil otherwise.

Getting a Web Archive

- [webArchive](#) (page 12)
Returns a web archive representing the receiver, its subresources and subframes.

Accessing Subresources

- [mainResource](#) (page 9)
Creates and returns a WebResource representing the receiver.

- [addSubresource](#): (page 7)
Adds a resource to the receiver's list of subresources.
- [subresourceForURL](#): (page 11)
Returns a subresource for the given URL.
- [subresources](#) (page 11)
Returns the receiver's subresources that have finished downloading.

Instance Methods

addSubresource:

Adds a resource to the receiver's list of subresources.

```
- (void)addSubresource:(WebResource *)subresource
```

Discussion

If the receiver needs to reload the resource's URL, it will load the data from *subresource* instead of the network. For example, use this method if you want to use a previously downloaded image rather than accessing the network to reload a resource. If the receiver already has a resource with the same URL as *subresource*, then this method replaces it.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [subresourceForURL](#): (page 11)
- [subresources](#) (page 11)

Declared In

WebDataSource.h

data

Returns the raw data that represents the receiver's content.

```
- (NSData *)data
```

Discussion

The format of the data is dependent on the receiver's MIME type (obtained from the response). The data will be incomplete until the data has finished loading. Returns *nil* if the receiver hasn't loaded any data. Use the [isLoading](#) (page 8) method to test if a data source is in the process of loading.

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

- [pageTitle](#) (page 9)
- [representation](#) (page 9)

Declared In

WebDataSource.h

initialRequest

Returns a reference to the original request that was used to load the web content.

- (NSURLRequest *)initialRequest

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

- [request](#) (page 10)

- [response](#) (page 10)

Declared In

WebDataSource.h

initWithRequest:

initializes a data source with a URL request.

- (id)initWithRequest:(NSURLRequest *)request

Discussion

This method is the designated initializer for WebDataSource objects where *request* is used to load the web content. Normally, WebFrame objects create their data sources, so don't invoke this method directly.

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

WebDataSource.h

isLoading

Returns YES if the receiver is in the process of loading its content, NO otherwise.

- (BOOL)isLoading

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

- [data](#) (page 7)

- [pageTitle](#) (page 9)

- [representation](#) (page 9)

Declared In

WebDataSource.h

mainResource

Creates and returns a WebResource representing the receiver.

- (WebResource *)mainResource

Discussion

The contents returned are based on the original downloaded data. You can use the returned value to create a WebArchive object instead of using the [webArchive](#) (page 12) method.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

WebDataSource.h

pageTitle

Returns the title of the receiver's page.

- (NSString *)pageTitle

Discussion

May return `nil` if the page has no title, or the page title hasn't been loaded yet. The WebView will notify its frame load delegate when the page title is loaded by invoking the `webView:didReceiveTitle:forFrame:delegate` method.

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

- [data](#) (page 7)
- [isLoading](#) (page 8)
- [representation](#) (page 9)

Related Sample Code

CarbonCocoaCoreImageTab

Declared In

WebDataSource.h

representation

Returns the receiver's representation depending on its MIME type.

- (id < WebDocumentRepresentation >)representation

Discussion

If the receiver is in the process of being loaded, this method may return `nil` if invoked before loading is complete. You can specify the mapping between a representation and MIME type using the `WebView.registerViewClass:representationClass:forMIMEType: class method`.

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

- [data](#) (page 7)
- [isLoading](#) (page 8)
- [pageTitle](#) (page 9)

Declared In

`WebDataSource.h`

request

Returns the request that was used to create the receiver.

- (`NSMutableURLRequest *`)request

Discussion

The URL returned may be different from the original request. A `WebView`'s resource load delegate may modify requests by implementing `webView:resource:willSendRequest:redirectResponse:fromDataSource:`.

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

- [initialRequest](#) (page 8)
- [response](#) (page 10)

Related Sample Code

`CarbonCocoaCoreImageTab`

Declared In

`WebDataSource.h`

response

Returns the associated `WebResourceResponse` for this data source.

- (`NSURLResponse *`)response

Discussion

This method returns `nil` if a response has not been received yet.

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

- [initialRequest](#) (page 8)
- [request](#) (page 10)

Declared In

WebDataSource.h

subresourceForURL:

Returns a subresource for the given URL.

- (WebResource *)subresourceForURL:(NSURL *)URL

Discussion

Return `nil` if the receiver hasn't finished downloading the subresource.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [addSubresource:](#) (page 7)
- [subresources](#) (page 11)

Declared In

WebDataSource.h

subresources

Returns the receiver's subresources that have finished downloading.

- (NSArray *)subresources

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [addSubresource:](#) (page 7)
- [subresourceForURL:](#) (page 11)

Declared In

WebDataSource.h

textEncodingName

Returns either the text encoding for the receiver's `WebView`, if set, or the text encoding of the response.

- (NSString *)textEncodingName

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

WebDataSource.h

unreachableURL

Returns the receiver's unreachable URL if it exists, `nil` otherwise.

- (NSURL *)unreachableURL

Discussion

The receiver will have an unreachable URL if it was created using the `loadAlternateHTMLString:baseURL:forUnreachableURL: WebFrame` method.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

WebDataSource.h

webArchive

Returns a web archive representing the receiver, its subresources and subframes.

- (WebArchive *)webArchive

Discussion

Constructs the web archive using the original downloaded data. In the case of HTML, if the current content is preferred, then send `webArchive` to the appropriate DOM object.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [mainResource](#) (page 9)

Declared In

WebDataSource.h

webFrame

Returns the web frame that represents this data source.

- (WebFrame *)webFrame

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

WebDataSource.h

Document Revision History

This table describes the changes to *WebDataSource Class Reference*.

Date	Notes
2008-10-15	Minor edits throughout.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

addSubresource: [instance method 7](#)

D

data [instance method 7](#)

I

initialRequest [instance method 8](#)
initWithRequest: [instance method 8](#)
isLoading [instance method 8](#)

M

mainResource [instance method 9](#)

P

pageTitle [instance method 9](#)

R

representation [instance method 9](#)
request [instance method 10](#)
response [instance method 10](#)

S

subresourceForURL: [instance method 11](#)
subresources [instance method 11](#)

T

textEncodingName [instance method 11](#)

U

unreachableURL [instance method 12](#)

W

webArchive [instance method 12](#)
webFrame [instance method 12](#)