

---

# UIWebView Class Reference

User Experience: Windows & Views



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

iPhone and Numbers are trademarks of Apple Inc.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. 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

## **UIWebView Class Reference 5**

---

Overview	5
Tasks	6
Setting the Delegate	6
Loading Content	6
Moving Back and Forward	6
Setting Web Content Properties	6
Running JavaScript	7
Properties	7
canGoBack	7
canGoForward	7
delegate	7
detectsPhoneNumbers	8
loading	8
request	9
scalesPageToFit	9
Instance Methods	9
goBack	9
goForward	10
loadData:MIMEType:textEncodingName:baseURL:	10
loadHTMLString:baseURL:	11
loadRequest:	11
reload	11
stopLoading	12
stringByEvaluatingJavaScriptFromString:	12
Constants	13
UIWebViewNavigationType	13

## **Document Revision History 15**

---

## **Index 17**

---

C O N T E N T S

# UIWebView Class Reference

---

<b>Inherits from</b>	UIView : UIResponder : NSObject
<b>Conforms to</b>	NSCoding NSCoding (UIView) NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/UIKit.framework
<b>Availability</b>	Available in iPhone OS 2.0 and later.
<b>Declared in</b>	UIWebView.h
<b>Related sample code</b>	UICatalog

## Overview

You use the `UIWebView` class to embed web content in your application. To do so, you simply create a `UIWebView` object, attach it to a window, and send it a request to load web content. You can also use this class to move back and forward in the history of webpages, and you can even set some web content properties programmatically.

Use the [loadRequest:](#) (page 11) method to begin loading web content, the [stopLoading](#) (page 12) method to stop loading, and the [loading](#) (page 8) property to find out if a web view is in the process of loading.

If you allow the user to move back and forward through the webpage history, then you can use the [goBack](#) (page 9) and [goForward](#) (page 10) methods as actions for buttons. Use the [canGoBack](#) (page 7) and [canGoForward](#) (page 7) properties to disable the buttons when the user can't move in a direction.

By default, a web view automatically converts telephone numbers that appear in web content to Phone links. When a Phone link is tapped, the Phone application launches and dials the number. Set the [detectsPhoneNumbers](#) (page 8) property to `NO` to turn off this default behavior.

You can also use the [request](#) (page 9) property to programmatically set the scale of web content the first time it is displayed in a web view. Thereafter, the user can change the scale using gestures.

Set the [delegate](#) (page 7) property to an object conforming to the `UIWebViewDelegate` protocol if you want to track the loading of web content.

Read *Safari Web Content Guide for iPhone OS* for how to create web content that is compatible with and optimized for displaying in Safari on iPhone and your web views.

## Tasks

### Setting the Delegate

[delegate](#) (page 7) *property*

The receiver's delegate.

### Loading Content

- [loadData:MIMETYPE:textEncodingName:baseURL:](#) (page 10)  
Sets the main page contents, MIME type, content encoding, and base URL.
- [loadHTMLString:baseURL:](#) (page 11)  
Sets the main page content and base URL.
- [loadRequest:](#) (page 11)  
Connects to a given URL by initiating an asynchronous client request.
- [request](#) (page 9) *property*  
The URL request identifying the location of the content to load. (read-only)
- [loading](#) (page 8) *property*  
A Boolean value indicating whether the receiver is done loading content. (read-only)
- [stopLoading](#) (page 12)  
Stops the loading of any web content managed by the receiver.
- [reload](#) (page 11)  
Reloads the current page.

### Moving Back and Forward

- [canGoBack](#) (page 7) *property*  
A Boolean value indicating whether the receiver can move backward. (read-only)
- [canGoForward](#) (page 7) *property*  
A Boolean value indicating whether the receiver can move forward. (read-only)
- [goBack](#) (page 9)  
Loads the previous location in the back-forward list.
- [goForward](#) (page 10)  
Loads the next location in the back-forward list.

### Setting Web Content Properties

[detectsPhoneNumbers](#) (page 8) *property*

A Boolean value indicating whether telephone number detection is on.

[scalesPageToFit](#) (page 9) *property*

A Boolean value determining whether the webpage scales to fit the view and the user can change the scale.

## Running JavaScript

- [stringByEvaluatingJavaScriptFromString:](#) (page 12)  
Returns the result of running a script.

## Properties

For more about Objective-C properties, see “Properties” in *The Objective-C 2.0 Programming Language*.

### canGoBack

A Boolean value indicating whether the receiver can move backward. (read-only)

```
@property(n nonatomic, readonly, getter=canGoBack) BOOL canGoBack
```

#### Discussion

If YES, able to move backward; otherwise, NO.

#### Availability

Available in iPhone OS 2.0 and later.

#### See Also

[@property canGoForward](#) (page 7)

#### Declared In

UIWebView.h

### canGoForward

A Boolean value indicating whether the receiver can move forward. (read-only)

```
@property(n nonatomic, readonly, getter=canGoForward) BOOL canGoForward
```

#### Discussion

If YES, able to move forward; otherwise, NO.

#### Availability

Available in iPhone OS 2.0 and later.

#### See Also

[@property canGoBack](#) (page 7)

#### Declared In

UIWebView.h

### delegate

The receiver’s delegate.

```
@property(n nonatomic, assign) id<UIWebViewDelegate> delegate
```

**Discussion**

The delegate is sent messages when content is loading. See *UIWebViewDelegate Protocol Reference* for the optional methods this delegate may implement.

**Important:** Before releasing an instance of UIWebView for which you have set a delegate, you must first set its delegate property to `nil`. This can be done, for example, in your dealloc method.

**Availability**

Available in iPhone OS 2.0 and later.

**Related Sample Code**

UICatalog

**Declared In**

UIWebView.h

**detectsPhoneNumbers**

A Boolean value indicating whether telephone number detection is on.

```
@property(n nonatomic) BOOL detectsPhoneNumbers
```

**Discussion**

If YES, telephone number detection is on; otherwise, NO. If a webpage contains numbers that can be interpreted as phone numbers, but are not phone numbers, you can turn off telephone number detection by setting this property to NO. The default value is YES on devices that have phone capabilities.

**Availability**

Available in iPhone OS 2.0 and later.

**Declared In**

UIWebView.h

**loading**

A Boolean value indicating whether the receiver is done loading content. (read-only)

```
@property(n nonatomic, readonly, getter=isLoading) BOOL loading
```

**Discussion**

If YES, the receiver is still loading content; otherwise, NO.

**Availability**

Available in iPhone OS 2.0 and later.

**See Also**

- [@property request](#) (page 9)
- [stopLoading](#) (page 12)
- [loadRequest:](#) (page 11)

- [reload](#) (page 11)

**Declared In**

UIWebView.h

**request**

The URL request identifying the location of the content to load. (read-only)

@property(nonatomic, readonly, retain) NSURLRequest \*request

**Availability**

Available in iPhone OS 2.0 and later.

**See Also**

- [loadRequest:](#) (page 11)
- [stopLoading](#) (page 12)
- [@property loading](#) (page 8)
- [reload](#) (page 11)

**Declared In**

UIWebView.h

**scalesPageToFit**

A Boolean value determining whether the webpage scales to fit the view and the user can change the scale.

@property(nonatomic) BOOL scalesPageToFit

**Discussion**

If YES, the webpage is scaled to fit and the user can zoom in and zoom out. If NO, user zooming is disabled. The default value is NO.

**Availability**

Available in iPhone OS 2.0 and later.

**Related Sample Code**

UICatalog

**Declared In**

UIWebView.h

## Instance Methods

**goBack**

Loads the previous location in the back-forward list.

- (void)goBack

**Availability**

Available in iPhone OS 2.0 and later.

**See Also**

- [goBack](#) (page 9)

**Declared In**

UIWebView.h

## goForward

Loads the next location in the back-forward list.

- (void)goForward

**Availability**

Available in iPhone OS 2.0 and later.

**See Also**

- [goBack](#) (page 9)

**Declared In**

UIWebView.h

## loadData:MIMEType:textEncodingName:baseURL:

Sets the main page contents, MIME type, content encoding, and base URL.

```
- (void)loadData:(NSData *)data MIMEType:(NSString *)MIMEType
    textEncodingName:(NSString *)encodingName baseURL:(NSURL *)baseURL
```

**Parameters**

*data*

The content for the main page.

*MIMEType*

The MIME type of the content.

*encodingName*

The IANA encoding name as in `utf-8` or `utf-16`.

*baseURL*

The base URL for the content.

**Availability**

Available in iPhone OS 2.0 and later.

**See Also**

- [loadHTMLString:baseURL:](#) (page 11)

**Declared In**

UIWebView.h

## loadHTMLString:baseURL:

Sets the main page content and base URL.

```
- (void)loadHTMLString:(NSString *)string baseURL:(NSURL *)baseURL
```

### Parameters

*string*

The content for the main page.

*baseURL*

The base URL for the content.

### Availability

Available in iPhone OS 2.0 and later.

### See Also

- [loadData:MIMETYPE:textEncodingName:baseURL:](#) (page 10)

### Declared In

UIWebView.h

## loadRequest:

Connects to a given URL by initiating an asynchronous client request.

```
- (void)loadRequest:(NSURLRequest *)request
```

### Parameters

*request*

A URL request identifying the location of the content to load.

### Discussion

To stop this load, use the [stopLoading](#) (page 12) method. To see whether the receiver is done loading the content, use the [loading](#) (page 8) property.

### Availability

Available in iPhone OS 2.0 and later.

### See Also

- [@property request](#) (page 9)
- [stopLoading](#) (page 12)
- [@property loading](#) (page 8)
- [reload](#) (page 11)

### Declared In

UIWebView.h

## reload

Reloads the current page.

```
- (void)reload
```

**Availability**

Available in iPhone OS 2.0 and later.

**See Also**

- [@property request](#) (page 9)
- [@property loading](#) (page 8)
- [loadRequest:](#) (page 11)
- [stopLoading](#) (page 12)

**Declared In**

UIWebView.h

**stopLoading**

Stops the loading of any web content managed by the receiver.

- (void)stopLoading

**Discussion**

Stops any content in the process of being loaded by the main frame or any of its children frames. Does nothing if no content is being loaded.

**Availability**

Available in iPhone OS 2.0 and later.

**See Also**

- [@property request](#) (page 9)
- [@property loading](#) (page 8)
- [loadRequest:](#) (page 11)
- [reload](#) (page 11)

**Declared In**

UIWebView.h

**stringByEvaluatingJavaScriptFromString:**

Returns the result of running a script.

- (NSString \*)stringByEvaluatingJavaScriptFromString:(NSString \*)*script*

**Parameters**

*script*

The script to run.

**Return Value**

The result of running *script* or *nil* if it fails.

**Discussion**

JavaScript execution time is limited to 5 seconds for each top-level entry point. If your script executes for more than 5 seconds, Safari stops executing the script. This is likely to occur at a random place in your code, so unintended consequences may result. This limit is imposed because JavaScript execution may cause the main thread to block, so when scripts are running, the user is not able to interact with the webpage.

JavaScript allocations are also limited to 10 MB. Safari raises an exception if you exceed this limit on the total memory allocation for JavaScript.

#### Availability

Available in iPhone OS 2.0 and later.

#### Declared In

UIWebView.h

## Constants

### UIWebViewNavigationType

Constant indicating the user's action.

```
enum {
    UIWebViewNavigationTypeLinkClicked,
    UIWebViewNavigationTypeFormSubmitted,
    UIWebViewNavigationTypeBackForward,
    UIWebViewNavigationTypeReload,
    UIWebViewNavigationTypeFormResubmitted,
    UIWebViewNavigationTypeOther
};
typedef NSInteger UIWebViewNavigationType;
```

#### Constants

UIWebViewNavigationTypeLinkClicked

User tapped a link.

Available in iPhone OS 2.0 and later.

Declared in UIWebView.h.

UIWebViewNavigationTypeFormSubmitted

User submitted a form.

Available in iPhone OS 2.0 and later.

Declared in UIWebView.h.

UIWebViewNavigationTypeBackForward

User tapped the back or forward button.

Available in iPhone OS 2.0 and later.

Declared in UIWebView.h.

UIWebViewNavigationTypeReload

User tapped the reload button.

Available in iPhone OS 2.0 and later.

Declared in UIWebView.h.

UIWebViewNavigationTypeFormResubmitted

User resubmitted a form.

Available in iPhone OS 2.0 and later.

Declared in UIWebView.h.

UIWebViewNavigationTypeOther

Some other action occurred.

Available in iPhone OS 2.0 and later.

Declared in `UIWebView.h`.

**Availability**

Available in iPhone OS 2.0 and later.

**Declared In**

`UIWebView.h`

# Document Revision History

---

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

Date	Notes
2008-10-15	Added important release information to delegate.
2008-09-09	Removed text about delegate being informed of scale changes.
2008-05-30	New document that describes the class for embedding web content in an application.

R E V I S I O N   H I S T O R Y

Document Revision History

# Index

---

## C

---

canGoBack **instance property** [7](#)  
canGoForward **instance property** [7](#)

## D

---

delegate **instance property** [7](#)  
detectsPhoneNumbers **instance property** [8](#)

## G

---

goBack **instance method** [9](#)  
goForward **instance method** [10](#)

## L

---

loadData:MIMETYPE:textEncodingName:baseURL:  
**instance method** [10](#)  
loadHTMLString:baseURL: **instance method** [11](#)  
loading **instance property** [8](#)  
loadRequest: **instance method** [11](#)

## R

---

reload **instance method** [11](#)  
request **instance property** [9](#)

## S

---

scalesPageToFit **instance property** [9](#)  
stopLoading **instance method** [12](#)

stringByEvaluatingJavaScriptFromString:  
**instance method** [12](#)

## U

---

UIWebViewNavigationType **data type** [13](#)  
UIWebViewNavigationTypeBackForward **constant** [13](#)  
UIWebViewNavigationTypeFormResubmitted  
**constant** [13](#)  
UIWebViewNavigationTypeFormSubmitted **constant**  
[13](#)  
UIWebViewNavigationTypeLinkClicked **constant** [13](#)  
UIWebViewNavigationTypeOther **constant** [14](#)  
UIWebViewNavigationTypeReload **constant** [13](#)