
NSMutableURLRequest Class Reference

[Cocoa](#) > [Networking](#)



2006-05-23



Apple Inc.
© 2006 Apple Computer, 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, Mac, Mac OS, and Safari are trademarks of Apple Inc., registered in the United States and other countries.

iPhone is a trademark of Apple Inc.

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

NSMutableURLRequest Class Reference 5

Overview	5
Tasks	5
Setting Request Properties	5
Setting HTTP Specific Properties	6
Instance Methods	6
addValue:forHTTPHeaderField:	6
setAllHTTPHeaderFields:	7
setCachePolicy:	7
setHTTPBody:	7
setHTTPBodyStream:	8
setHTTPMethod:	8
setHTTPShouldHandleCookies:	9
setMainDocumentURL:	9
setTimeoutInterval:	9
setURL:	10
setValue:forHTTPHeaderField:	10

Document Revision History 13

Index 15

NSMutableURLRequest Class Reference

Inherits from	NSURLRequest : NSObject
Conforms to	NSCoding (NSURLRequest) NSCopying (NSURLRequest) NSMutableCopying (NSURLRequest) NSObject (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	NSURLRequest.h

Overview

NSMutableURLRequest is a subclass of NSURLRequest provided to aid developers who may find it more convenient to mutate a single request object for a series of URL load requests instead of creating an immutable NSURLRequest for each load.

This programming model is supported by the following contract between NSMutableURLRequest and NSURLConnection: NSURLConnection makes a deep copy of each NSMutableURLRequest object passed to one of its initializers.

NSMutableURLRequest, like NSURLRequest, is designed to be extended to support additional protocols by adding categories that access protocol specific values from a property object using NSURLProtocol's `propertyForKey:inRequest:` and `setProperty:forKey:inRequest:` methods.

Tasks

Setting Request Properties

- [setCachePolicy:](#) (page 7)
Sets the cache policy of the receiver.
- [setMainDocumentURL:](#) (page 9)
Sets the main document URL for the receiver.

- [setTimeoutInterval:](#) (page 9)
Sets the receiver's timeout interval, in seconds.
- [setURL:](#) (page 10)
Sets the URL of the receiver

Setting HTTP Specific Properties

- [addValue:forHTTPHeaderField:](#) (page 6)
Adds an HTTP header to the receiver's HTTP header dictionary.
- [setAllHTTPHeaderFields:](#) (page 7)
Replaces the receiver's header fields with the passed values.
- [setHTTPBody:](#) (page 7)
Sets the request body of the receiver to the specified data.
- [setHTTPBodyStream:](#) (page 8)
Sets the request body of the receiver to the contents of a specified input stream.
- [setHTTPMethod:](#) (page 8)
Sets the receiver's HTTP request method.
- [setHTTPShouldHandleCookies:](#) (page 9)
Sets whether the receiver should use the default cookie handling for the request.
- [setValue:forHTTPHeaderField:](#) (page 10)
Sets the specified HTTP header field.

Instance Methods

addValue:forHTTPHeaderField:

Adds an HTTP header to the receiver's HTTP header dictionary.

```
- (void)addValue:(NSString *)value forHTTPHeaderField:(NSString *)field
```

Parameters

value

The value for the header field.

field

The name of the header field. In keeping with the HTTP RFC, HTTP header field names are case-insensitive

Discussion

This method provides the ability to add values to header fields incrementally. If a value was previously set for the specified *field*, the supplied *value* is appended to the existing value using the appropriate field delimiter. In the case of HTTP, this is a comma.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

- [setValue:forHTTPHeaderField:](#) (page 10)

Declared In

NSURLRequest.h

setAllHTTPHeaderFields:

Replaces the receiver's header fields with the passed values.

```
- (void)setAllHTTPHeaderFields:(NSDictionary *)headerFields
```

Parameters

headerFields

A dictionary with the new header fields. HTTP header fields must be string values; therefore, each object and key in the *headerFields* dictionary must be a subclass of NSString. If either the key or value for a key-value pair is not a subclass of NSString, the key-value pair is skipped.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

- [setValue:forHTTPHeaderField:](#) (page 10)

Declared In

NSURLRequest.h

setCachePolicy:

Sets the cache policy of the receiver.

```
- (void)setCachePolicy:(NSURLRequestCachePolicy)policy
```

Parameters

policy

The new cache policy.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

- `cachePolicy`

Declared In

NSURLRequest.h

setHTTPBody:

Sets the request body of the receiver to the specified data.

```
- (void)setHTTPBody:(NSData *)data
```

Parameters

data

The new request body for the receiver. This is sent as the message body of the request, as in an HTTP POST request.

Discussion

Setting the HTTP body data clears any input stream set by [setHTTPBodyStream:](#) (page 8). These values are mutually exclusive.

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

NSURLRequest.h

setHTTPBodyStream:

Sets the request body of the receiver to the contents of a specified input stream.

```
- (void)setHTTPBodyStream:(NSInputStream *)inputStream
```

Parameters

inputStream

The input stream that will be the request body of the receiver. The entire contents of the stream will be sent as the body, as in an HTTP POST request. The *inputStream* should be unopened and the receiver will take over as the stream's delegate.

Discussion

Setting a body stream clears any data set by [setHTTPBody:](#) (page 7). These values are mutually exclusive.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSURLRequest.h

setHTTPMethod:

Sets the receiver's HTTP request method.

```
- (void)setHTTPMethod:(NSString *)method
```

Parameters

method

The new HTTP request method. The default HTTP method is "GET".

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

NSURLRequest.h

setHTTPShouldHandleCookies:

Sets whether the receiver should use the default cookie handling for the request.

```
- (void)setHTTPShouldHandleCookies:(BOOL)handleCookies
```

Parameters

handleCookies

YES if the receiver should use the default cookie handling for the request, NO otherwise. The default is YES.

Special Considerations

In Mac OS X v10.2 with Safari 1.0 the value set by this method is not respected by the framework.

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

NSURLRequest.h

setMainDocumentURL:

Sets the main document URL for the receiver.

```
- (void)setMainDocumentURL:(NSURL *)theURL
```

Parameters

theURL

The new main document URL. Can be nil.

Discussion

The caller should set the main document URL to an appropriate main document, if known. For example, when loading a web page the URL of the HTML document for the top-level frame would be appropriate. This URL will be used for the “only from same domain as main document” cookie accept policy.

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

NSURLRequest.h

setTimeoutInterval:

Sets the receiver's timeout interval, in seconds.

```
- (void)setTimeoutInterval:(NSTimeInterval)timeoutInterval
```

Parameters*timeoutInterval*

The timeout interval, in seconds. If during a connection attempt the request remains idle for longer than the timeout interval, the request is considered to have timed out. The default timeout interval is 60 seconds.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

- `timeoutInterval`

Declared In

NSURLRequest.h

setURL:

Sets the URL of the receiver

- (void)setURL:(NSURL *)*theURL*

Parameters*theURL*

The new URL.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

- URL

Declared In

NSURLRequest.h

setValue:forHTTPHeaderField:

Sets the specified HTTP header field.

- (void)setValue:(NSString *)*value* forHTTPHeaderField:(NSString *)*field*

Parameters*value*

The new value for the header field. Any existing value for the field is replaced by the new value.

field

The name of the header field to set. In keeping with the HTTP RFC, HTTP header field names are case-insensitive.

Availability

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

Available in Mac OS X v10.2.7 and later.

See Also

- [addValue:forHTTPHeaderField:](#) (page 6)

Declared In

NSURLRequest.h

Document Revision History

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

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

REVISION HISTORY

Document Revision History

Index

A

`addValue:forHTTPHeaderField:` [instance method 6](#)

S

`setAllHTTPHeaderFields:` [instance method 7](#)

`setCachePolicy:` [instance method 7](#)

`setHTTPBody:` [instance method 7](#)

`setHTTPBodyStream:` [instance method 8](#)

`setHTTPMethod:` [instance method 8](#)

`setHTTPShouldHandleCookies:` [instance method 9](#)

`setMainDocumentURL:` [instance method 9](#)

`setTimeoutInterval:` [instance method 9](#)

`setURL:` [instance method 10](#)

`setValue:forHTTPHeaderField:` [instance method 10](#)