
NSURLAuthenticationChallenge Class Reference

Networking, Internet, & Web: Protocol Streams



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

NSURLAuthenticationChallenge Class Reference 5

Overview 5

Tasks 5

 Creating an Authentication Challenge Instance 5

 Getting Authentication Challenge Properties 5

Instance Methods 6

 error 6

 failureResponse 6

 initWithAuthenticationChallenge:sender: 7

 initWithProtectionSpace:proposedCredential:previousFailureCount:failureResponse: error:sender:
7

 previousFailureCount 8

 proposedCredential 8

 protectionSpace 8

 sender 8

Document Revision History 11

Index 13

NSURLAuthenticationChallenge Class Reference

Inherits from	NSObject
Conforms to	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	NSURLAuthenticationChallenge.h
Related sample code	ImageClient LSMSmartCategorizer

Overview

NSURLAuthenticationChallenge encapsulates a challenge from a server requiring authentication from the client.

Tasks

Creating an Authentication Challenge Instance

- [initWithAuthenticationChallenge:sender:](#) (page 7)
Returns an initialized NSURLAuthenticationChallenge object copying the properties from *challenge*, and setting the authentication sender to *sender*.
- [initWithProtectionSpace:proposedCredential:previousFailureCount:failureResponse:error:sender:](#) (page 7)
Returns an initialized NSURLAuthenticationChallenge object for the specified *space* using the *credential*, or *nil* if there is no proposed credential.

Getting Authentication Challenge Properties

- [error](#) (page 6)
Returns the NSError object representing the last authentication failure.

- [failureResponse](#) (page 6)
Returns the NSURLResponse object representing the last authentication failure.
- [previousFailureCount](#) (page 8)
Returns the receiver's count of failed authentication attempts.
- [proposedCredential](#) (page 8)
Returns the proposed credential for this challenge.
- [protectionSpace](#) (page 8)
Returns the receiver's protection space.
- [sender](#) (page 8)
Returns the receiver's sender.

Instance Methods

error

Returns the NSError object representing the last authentication failure.

- (NSError *)error

Discussion

This method returns `nil` if the protocol doesn't use errors to indicate an authentication failure.

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

- [failureResponse](#) (page 6)

Declared In

NSURLAuthenticationChallenge.h

failureResponse

Returns the NSURLResponse object representing the last authentication failure.

- (NSURLResponse *)failureResponse

Discussion

This method will return `nil` if the protocol doesn't use responses to indicate an authentication failure.

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

- [error](#) (page 6)

Declared In

NSURLAuthenticationChallenge.h

initWithAuthenticationChallenge:sender:

Returns an initialized NSURLAuthenticationChallenge object copying the properties from *challenge*, and setting the authentication sender to *sender*.

```
- (id)initWithAuthenticationChallenge:(NSURLAuthenticationChallenge *)challenge
    sender:(id < NSURLAuthenticationChallengeSender >)sender
```

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

- [initWithProtectionSpace:proposedCredential:previousFailureCount:failureResponse:error:sender:](#) (page 7)

Declared In

NSURLAuthenticationChallenge.h

initWithProtectionSpace:proposedCredential:previousFailureCount:failureResponse:error:sender:

Returns an initialized NSURLAuthenticationChallenge object for the specified *space* using the *credential*, or *nil* if there is no proposed credential.

```
- (id)initWithProtectionSpace:(NSURLProtectionSpace *)space
    proposedCredential:(NSURLCredential *)credential
    previousFailureCount:(NSInteger)count failureResponse:(NSURLResponse *)response
    error:(NSError *)error sender:(id < NSURLAuthenticationChallengeSender >)sender
```

Discussion

The previous failure count is set to *count*. The *response* should contain the NSURLResponse for the authentication failure, or *nil* if it is not applicable to the challenge. The *error* should contain the NSError for the authentication failure, or *nil* if it is not applicable to the challenge. The object that initiated the authentication challenge is set to *sender*.

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

- [initWithAuthenticationChallenge:sender:](#) (page 7)

Declared In

NSURLAuthenticationChallenge.h

previousFailureCount

Returns the receiver's count of failed authentication attempts.

- (NSInteger)previousFailureCount

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

NSURLAuthenticationChallenge.h

proposedCredential

Returns the proposed credential for this challenge.

- (NSURLCredential *)proposedCredential

Discussion

This method will return `nil` if there is no default credential for this challenge.

If the proposed credential is not `nil` and returns YES when sent the message `hasPassword`, then the credential is ready to use as-is. If the proposed credential returns NO for `hasPassword`, then the credential provides a default user name and the client must prompt the user for a corresponding password.

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

NSURLAuthenticationChallenge.h

protectionSpace

Returns the receiver's protection space.

- (NSURLProtectionSpace *)protectionSpace

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

NSURLAuthenticationChallenge.h

sender

Returns the receiver's sender.

- (id < NSURLAuthenticationChallengeSender >)sender

Discussion

The sender should be sent a `useCredential:forAuthenticationChallenge:`, `continueWithoutCredentialForAuthenticationChallenge:` or `cancelAuthenticationChallenge:` when the client is finished processing the authentication challenge.

Availability

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

Available in Mac OS X v10.2.7 and later.

Related Sample Code

`ImageClient`

Declared In

`NSURLAuthenticationChallenge.h`

Document Revision History

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

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

REVISION HISTORY

Document Revision History

Index

E

error [instance method 6](#)

F

failureResponse [instance method 6](#)

I

initWithAuthenticationChallenge:sender:
[instance method 7](#)

initWithProtectionSpace:proposedCredential:
previousFailureCount:failureResponse:error:sender:
[instance method 7](#)

P

previousFailureCount [instance method 8](#)

proposedCredential [instance method 8](#)

protectionSpace [instance method 8](#)

S

sender [instance method 8](#)