CFHTTPAuthentication Reference

Core Foundation > Networking

2007-05-15

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

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

CFHTTPAuthentication Reference

Derived From:	CFType
Framework:	CoreServices
Companion guide	CFNetwork Programming Guide
Declared in	CFHTTPAuthentication.h CFHTTPMessage.h

Overview

The CFHTTPAuthentication opaque type provides an abstraction of HTTP authentication information.

Functions by Task

Creating an HTTP authentication

`CFHTTPAuthenticationCreateFromResponse` (page 8)
Uses an authentication failure response to create a CFHTTPAuthentication object.

CFHTTP Authentication Functions

This section describes the CFNetwork authentication functions that are used to manage authentication information associated with a request. The functions work with a CFHTTPAuthentication object, which is created from an HTTP response that failed with a 401 or 407 error code.

When you have analyzed the CFHTTPAuthentication object and acquired the necessary credentials to perform the authentication, call `CFHTTPMessageApplyCredentials` or `CFHTTPMessageApplyCredentialDictionary` to perform the authentication.

`CFHTTPAuthenticationAppliesToRequest` (page 6)
Returns a Boolean value that indicates whether a CFHTTPAuthentication object is associated with a CFHTTPMessage object.

`CFHTTPAuthenticationCopyDomains` (page 7)
Returns an array of domain URLs to which a given CFHTTPAuthentication object can be applied.

`CFHTTPAuthenticationCopyMethod` (page 7)
Gets the strongest authentication method that will be used when a CFHTTPAuthentication object is applied to a request.

Getting the CFHTTPAuthentication type ID

Functions

CFHTTPAuthenticationAppliesToRequest

Returns a Boolean value that indicates whether a CFHTTPAuthentication object is associated with a CFHTTPMessage object.

```
Boolean CFHTTPAuthenticationAppliesToRequest (
   CFHTTPAuthenticationRef auth,
   CFHTTPMessageRef request
);
```

Parameters

`auth`
: The CFHTTPAuthentication object to examine.

`request`
: Request that `auth` is to be tested against.

Return Value

`TRUE` if `auth` is associated with `request`, otherwise `FALSE`.

Discussion

If this function returns `TRUE`, you can use *`auth`* to provide authentication information when using *`request`*.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`CFHTTPAuthentication.h`

CFHTTPAuthenticationCopyDomains

Returns an array of domain URLs to which a given CFHTTPAuthentication object can be applied.

```
CFArrayRef CFHTTPAuthenticationCopyDomains (
   CFHTTPAuthenticationRef auth
);
```

Parameters

auth

The CFHTTPAuthentication object to examine.

Return Value

A CFArray object that contains the domain URL's to which `auth` should be applied. Ownership follows the Create Rule.

Discussion

This function is provided for informational purposes only.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CFHTTPAuthentication.h`

CFHTTPAuthenticationCopyMethod

Gets the strongest authentication method that will be used when a CFHTTPAuthentication object is applied to a request.

```
CFStringRef CFHTTPAuthenticationCopyMethod (
   CFHTTPAuthenticationRef auth
);
```

Parameters

auth

The CFHTTPAuthentication object to examine.

Return Value

A string containing the authentication method that will be used *auth* is applied to a request. If more than one authentication method is available, the strongest authentication method is returned. Ownership follows the Create Rule.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CFHTTPAuthentication.h`

CFHTTPAuthenticationCopyRealm

Gets an authentication information's namespace.

```
CFStringRef CFHTTPAuthenticationCopyRealm (
   CFHTTPAuthenticationRef auth
);
```

Parameters

auth

The CFHTTPAuthentication object to examine.

Return Value

The namespace, if there is one; otherwise `NULL`. Ownership follows the Create Rule.

Discussion

Some authentication methods provide a namespace, and it is usually used to prompt the user for a name and password.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CFHTTPAuthentication.h`

CFHTTPAuthenticationCreateFromResponse

Uses an authentication failure response to create a CFHTTPAuthentication object.

```
CFHTTPAuthenticationRef CFHTTPAuthenticationCreateFromResponse (
   CFAllocatorRef alloc,
   CFHTTPMessageRef response
);
```

Parameters

alloc

The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

response

Response indicating an authentication failure; usually a 401 or a 407 response.

Return Value

CFHTTPAuthentication object that can be used for adding credentials to future requests. Ownership follows the Create Rule.

Discussion

This function uses a response containing authentication failure information to create a reference to a CFHTTPAuthentication object. You can use the object to add credentials to future requests. You can query the object to get the following information:

- whether it can be used and re-used to authenticate with its corresponding server [`CFHTTPAuthenticationIsValid` (page 9)]
- the authentication method that will be used when it is used to perform an authentication [`CFHTTPAuthenticationCopyMethod` (page 7)]
- whether it is associated with a particular CFHTTPMessageRef [`CFHTTPAuthenticationAppliesToRequest` (page 6)

- whether a user name and a password will be required when it is used to perform an authentication [CFHTTPAuthenticationRequiresUserNameAndPassword (page 11)]
- whether an account domain will be required when it is used to perform an authentication [CFHTTPAuthenticationRequiresAccountDomain (page 10)]
- whether authentication requests should be sent one at a time to the corresponding server [CFHTTPAuthenticationRequiresOrderedRequests (page 10)]
- the namespace (if any) that the domain uses to prompt for a name and password [CFHTTPAuthenticationCopyRealm (page 7)]
- the domain URLs the instance applies to [CFHTTPAuthenticationCopyDomains (page 7)]

When you have determined what information will be needed to perform the authentication and accumulated that information, call CFHTTPMessageApplyCredentials or CFHTTPMessageApplyCredentialDictionary to perform the authentication.

Availability
Available in Mac OS X version 10.4 and later.

Declared In
CFHTTPAuthentication.h

CFHTTPAuthenticationGetTypeID

Gets the Core Foundation type identifier for the CFHTTPAuthentication opaque type.

```
CFTypeID CFHTTPAuthenticationGetTypeID ();
```

Return Value
The Core Foundation type identifier for the CFHTTPAuthentication opaque type.

Availability
Available in Mac OS X version 10.4 and later.

Declared In
CFHTTPAuthentication.h

CFHTTPAuthenticationIsValid

Returns a Boolean value that indicates whether a CFHTTPAuthentication object is valid.

```
Boolean CFHTTPAuthenticationIsValid (
   CFHTTPAuthenticationRef auth,
   CFStreamError *error
);
```

Parameters

auth

The CFHTTPAuthentication object to examine.

error

Pointer to a CFStreamError structure, whose fields, if an error has occurred, are set to the error and the error's domain.

Return Value
TRUE if auth contains enough information to be applied to a request.

If this function returns FALSE, the CFHTTPAuthentication object may still contain useful information, such as the name of an unsupported authentication method.

Discussion
If this function returns TRUE for *auth*, the object is good for use with functions such as CFHTTPMessageApplyCredentials and CFHTTPMessageApplyCredentialDictionary. If this function returns FALSE, *auth* is invalid, and authentications using it will not succeed.

Availability
Available in Mac OS X version 10.4 and later.

Declared In
CFHTTPAuthentication.h

CFHTTPAuthenticationRequiresAccountDomain

Returns a Boolean value that indicates whether a CFHTTPAuthentication object uses an authentication method that requires an account domain.

```
Boolean CFHTTPAuthenticationRequiresAccountDomain (
   CFHTTPAuthenticationRef auth
);
```

Parameters

auth

The CFHTTPAuthentication object to examine.

Return Value
TRUE if auth uses an authentication method that requires an account domain, otherwise FALSE.

Availability
Available in Mac OS X version 10.4 and later.

Declared In
CFHTTPAuthentication.h

CFHTTPAuthenticationRequiresOrderedRequests

Returns a Boolean value that indicates whether authentication requests should be made one at a time.

```
Boolean CFHTTPAuthenticationRequiresOrderedRequests (
   CFHTTPAuthenticationRef auth
);
```

Parameters

auth

The CFHTTPAuthentication object to examine.

Return Value
TRUE if auth requires ordered requests, otherwise FALSE.

Discussion

Some authentication methods require that future requests must be performed in an ordered manner. If this function returns `TRUE`, clients can improve their chances of authenticating successfully by issuing requests one at a time as responses come back from the server.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CFHTTPAuthentication.h`

CFHTTPAuthenticationRequiresUserNameAndPassword

Returns a Boolean value that indicates whether a CFHTTPAuthentication object uses an authentication method that requires a username and a password.

```
Boolean CFHTTPAuthenticationRequiresUserNameAndPassword (
   CFHTTPAuthenticationRef auth
);
```

Parameters

`auth`

The CFHTTPAuthentication object to examine.

Return Value

`TRUE` if `auth` requires a username and password when it is applied to a request; otherwise, `FALSE`.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CFHTTPAuthentication.h`

Data Types

CFHTTPAuthenticationRef

An opaque reference representing HTTP authentication information.

```
typedef struct __CFHTTPAuthentication *CFHTTPAuthenticationRef;
```

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CFHTTPAuthentication.h`

Constants

CFHTTP Authentication Scheme Constants

Specifies the authentication scheme when adding authentication information to a CFHTTP request message object.

```
const CFStringRef kCFHTTPAuthenticationSchemeBasic;
const CFStringRef kCFHTTPAuthenticationSchemeDigest;
const CFStringRef kCFHTTPAuthenticationSchemeNegotiate;
const CFStringRef kCFHTTPAuthenticationSchemeNTLM;
```

Constants

`kCFHTTPAuthenticationSchemeBasic`

Specifies basic authentication consisting of a user name and a password.

Available in Mac OS X v10.1 and later.

Declared in `CFHTTPMessage.h`.

`kCFHTTPAuthenticationSchemeDigest`

Reserved.

Available in Mac OS X v10.1 and later.

Declared in `CFHTTPMessage.h`.

`kCFHTTPAuthenticationSchemeNegotiate`

Specifies the Negotiate authentication scheme.

Available in Mac OS X v10.5 and later.

Declared in `CFHTTPMessage.h`.

`kCFHTTPAuthenticationSchemeNTLM`

Specifies the NTLM authentication scheme.

Available in Mac OS X v10.5 and later.

Declared in `CFHTTPMessage.h`.

Discussion

The authentication scheme constants are used to specify the authentication scheme when calling `CFHTTPMessageAddAuthentication`.

CFStream HTTP Authentication Error Constants

Authentication error codes that may be returned when trying to apply authentication to a request.

```
enum CFStreamErrorHTTPAuthentication{
    kCFStreamErrorHTTPAuthenticationTypeUnsupported = -1000,
    kCFStreamErrorHTTPAuthenticationBadUserName = -1001,
    kCFStreamErrorHTTPAuthenticationBadPassword = -1002
};
typedef enum CFStreamErrorHTTPAuthentication CFStreamErrorHTTPAuthentication;
```

Constants

`kCFStreamErrorHTTPAuthenticationTypeUnsupported`

Specified authentication type is not supported.

Available in Mac OS X v10.4 and later.

Declared in `CFHTTPAuthentication.h`.

`kCFStreamErrorHTTPAuthenticationBadUserName`

User name is in a format that is not suitable for the request. Currently, user names are decoded using `kCFStringEncodingISOLatin1`.

Available in Mac OS X v10.4 and later.

Declared in `CFHTTPAuthentication.h`.

`kCFStreamErrorHTTPAuthenticationBadPassword`

Password is in a format that is not suitable for the request. Currently, passwords are decoded using `kCFStringEncodingISOLatin1`.

Available in Mac OS X v10.4 and later.

Declared in `CFHTTPAuthentication.h`.

CFHTTPMessageApplyCredentialDictionary Keys

Constants for keys in the dictionary passed to `CFHTTPMessageApplyCredentialDictionary`.

```
const CFStringRef kCFHTTPAuthenticationUserName;
const CFStringRef kCFHTTPAuthenticationPassword;
const CFStringRef kCFHTTPAuthenticationAccountDomain;
```

Constants

`kCFHTTPAuthenticationUserName`

Username to use for authentication.

`kCFHTTPAuthenticationPassword`

Password to use for authentication.

Available in Mac OS X v10.4 and later.

Declared in `CFHTTPAuthentication.h`.

`kCFHTTPAuthenticationAccountDomain`

Account domain to use for authentication.

Available in Mac OS X v10.4 and later.

Declared in `CFHTTPAuthentication.h`.

Document Revision History

This table describes the changes to *CFHTTPAuthentication Reference*.

Date	Notes
2007-05-15	Made minor formatting changes.
2006-02-07	New document that describes the C API for applying authentication credentials to HTTP messages.

Index