NSURLProtocol Class Reference

Cocoa > Networking

2007-04-01

Contents

NSURLProtocol 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	NSURLProtocol.h

Overview

`NSURLProtocol` is an abstract class that provides the basic structure for performing protocol-specific loading of URL data. Concrete subclasses handle the specifics associated with one or more protocols or URL schemes.

An application should never need to directly instantiate an `NSURLProtocol` subclass. The instance of the appropriate `NSURLProtocol` subclass for an `NSURLRequest` is created by `NSURLConnection` when a download is started.

The `NSURLProtocolClient` protocol describes the methods an implementation uses to drive the URL loading system from a `NSURLProtocol` subclass.

To support customization of protocol-specific requests, protocol implementors are encouraged to provide categories on `NSURLRequest` and `NSMutableURLRequest`. Protocol implementors who need to extend the capabilities of `NSURLRequest` and `NSMutableURLRequest` in this way can store and retrieve protocol-specific request data by using `NSURLProtocol`'s class methods `propertyForKey:inRequest:` (page 8) and `setProperty:forKey:inRequest:` (page 10).

An essential responsibility for a protocol implementor is creating a `NSURLResponse` for each request it processes successfully. A protocol implementor may wish to create a custom, mutable `NSURLResponse` class to provide protocol specific information.

Tasks

Creating Protocol Objects

`- initWithRequest:cachedResponse:client:` (page 12)
Initializes an `NSURLProtocol` object.

Registering and Unregistering Protocol Classes

`+ registerClass:` (page 9)
Attempts to register a subclass of `NSURLProtocol`, making it visible to the URL loading system.

`+ unregisterClass:` (page 11)
Unregisters the specified subclass of `NSURLProtocol`.

Getting and Setting Request Properties

`+ propertyForKey:inRequest:` (page 8)
Returns the property associated with the specified key in the specified request.

`+ setProperty:forKey:inRequest:` (page 10)
Sets the property associated with the specified key in the specified request.

`+ removePropertyForKey:inRequest:` (page 9)
Removes the property associated with the specified key in the specified request.

Determining If a Subclass Can Handle a Request

`+ canInitWithRequest:` (page 7)
Returns whether the protocol subclass can handle the specified request.

Providing a Canonical Version of a Request

`+ canonicalRequestForRequest:` (page 7)
Returns a canonical version of the specified request.

Determining If Requests Are Cache Equivalent

`+ requestIsCacheEquivalent:toRequest:` (page 10)
Returns whether two requests are equivalent for cache purposes.

Starting and Stopping Downloads

`– startLoading` (page 13)
    Starts protocol-specific loading of the request.

`– stopLoading` (page 13)
    Stops protocol-specific loading of the request.

Getting Protocol Attributes

`– cachedResponse` (page 11)
    Returns the receiver's cached response.

`– client` (page 12)
    Returns the object the receiver uses to communicate with the URL loading system.

`– request` (page 12)
    Returns the receiver's request.

Class Methods

canInitWithRequest:

Returns whether the protocol subclass can handle the specified request.

```
+ (BOOL)canInitWithRequest:(NSURLRequest *)request
```

Parameters

`request`
    The request to be handled.

Return Value
`YES` if the protocol subclass can handle *`request`*, otherwise `NO`.

Discussion
A subclass should inspect *`request`* and determine whether or not the implementation can perform a load with that request.

This is an abstract method and subclasses must provide an implementation.

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
`NSURLProtocol.h`

canonicalRequestForRequest:

Returns a canonical version of the specified request.

```
+ (NSURLRequest *)canonicalRequestForRequest:(NSURLRequest *)request
```

Parameters

request

The request whose canonical version is desired.

Return Value

The canonical form of *request*.

Discussion

It is up to each concrete protocol implementation to define what “canonical” means. A protocol should guarantee that the same input request always yields the same canonical form.

Special consideration should be given when implementing this method, because the canonical form of a request is used to lookup objects in the URL cache, a process which performs equality checks between `NSURLRequest` objects.

This is an abstract method and subclasses must provide an implementation.

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

`NSURLProtocol.h`

propertyForKey:inRequest:

Returns the property associated with the specified key in the specified request.

```
+ (id)propertyForKey:(NSString *)key inRequest:(NSURLRequest *)request
```

Parameters

key

The key of the desired property.

request

The request whose properties are to be queried.

Return Value

The property associated with *key*, or `nil` if no property has been stored for *key*.

Discussion

This method provides an interface for protocol implementors to access protocol-specific information associated with `NSURLRequest` objects.

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

`+ setProperty:forKey:inRequest:` (page 10)

Declared In

`NSURLProtocol.h`

registerClass:

Attempts to register a subclass of `NSURLProtocol`, making it visible to the URL loading system.

```
+ (BOOL)registerClass:(Class)protocolClass
```

Parameters

protocolClass

The subclass of NSURLProtocol to register.

Return Value

`YES` if the registration is successful, `NO` otherwise. The only failure condition is if *protocolClass* is not a subclass of NSURLProtocol.

Discussion

When the URL loading system begins to load a request, each registered protocol class is consulted in turn to see if it can be initialized with the specified request. The first `NSURLProtocol` subclass to return `YES` when sent a `canInitWithRequest:` (page 7) message is used to perform the URL load. There is no guarantee that all registered protocol classes will be consulted.

Classes are consulted in the reverse order of their registration. A similar design governs the process to create the canonical form of a request with `canonicalRequestForRequest:` (page 7).

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

`+ unregisterClass:` (page 11)

Declared In

`NSURLProtocol.h`

removePropertyForKey:inRequest:

Removes the property associated with the specified key in the specified request.

```
+ (void)removePropertyForKey:((NSString *)key inRequest:(NSMutableURLRequest
    *)request
```

Parameters

key

The key whose value should be removed.

request

The request from which to remove the property value.

Discussion

This method is used to provide an interface for protocol implementors to customize protocol-specific information associated with `NSMutableURLRequest` objects.

Availability

Available in Mac OS X v10.5 and later.

See Also

`+ propertyForKey:inRequest:` (page 8)

Declared In
`NSURLProtocol.h`

requestIsCacheEquivalent:toRequest:

Returns whether two requests are equivalent for cache purposes.

```
+ (BOOL)requestIsCacheEquivalent:(NSURLRequest *)aRequest toRequest:(NSURLRequest
    *)bRequest
```

Parameters

aRequest
The request to compare with *bRequest*.

bRequest
The request to compare with *aRequest*.

Return Value
`YES` if *aRequest* and *bRequest* are equivalent for cache purposes, `NO` otherwise. Requests are considered equivalent for cache purposes if and only if they would be handled by the same protocol and that protocol declares them equivalent after performing implementation-specific checks.

Discussion
The `NSURLProtocol` implementation of this method compares the URLs of the requests to determine if the requests should be considered equivalent. Subclasses can override this method to provide protocol-specific comparisons.

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
`NSURLProtocol.h`

setProperty:forKey:inRequest:

Sets the property associated with the specified key in the specified request.

```
+ (void)setProperty:(id)value forKey:(NSString *)key inRequest:(NSMutableURLRequest
    *)request
```

Parameters

value
The value to set for the specified property.

key
The key for the specified property.

request
The request for which to create the property.

Discussion
This method is used to provide an interface for protocol implementors to customize protocol-specific information associated with `NSMutableURLRequest` objects.

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
`+ propertyForKey:inRequest:` (page 8)

Declared In
`NSURLProtocol.h`

unregisterClass:

Unregisters the specified subclass of `NSURLProtocol`.

```
+ (void)unregisterClass:(Class)protocolClass
```

Parameters

protocolClass
The subclass of NSURLProtocol to unregister.

Discussion
After this method is invoked, *protocolClass* is no longer consulted by the URL loading system.

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
`+ registerClass:` (page 9)

Declared In
`NSURLProtocol.h`

Instance Methods

cachedResponse

Returns the receiver's cached response.

```
- (NSCachedURLResponse *)cachedResponse
```

Return Value
The receiver's cached response.

Discussion
Subclasses must implement this method.

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
`NSURLProtocol.h`

client

Returns the object the receiver uses to communicate with the URL loading system.

```
- (id < NSURLProtocolClient >)client
```

Return Value
The object the receiver uses to communicate with the URL loading system.

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
`NSURLProtocol.h`

initWithRequest:cachedResponse:client:

Initializes an `NSURLProtocol` object.

```
- (id)initWithRequest:(NSURLRequest *)request cachedResponse:(NSCachedURLResponse
     *)cachedResponse client:(id < NSURLProtocolClient >)client
```

Parameters

request
The URL request for the URL protocol object.

cachedResponse
A cached response for the request; may be `nil` if there is no existing cached response for the request.

client
An object that provides an implementation of the NSURLProtocolClient protocol that the receiver uses to communicate with the URL loading system.

Discussion
Subclasses should override this method to do any custom initialization. An application should never explicitly call this method.

This is the designated intializer for `NSURLProtocol`.

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
`NSURLProtocol.h`

request

Returns the receiver's request.

```
- (NSURLRequest *)request
```

Return Value
The receiver's request.

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
`NSURLProtocol.h`

startLoading

Starts protocol-specific loading of the request.

```
- (void)startLoading
```

Discussion
When this method is called, the subclass implementation should start loading the request, providing feedback to the URL loading system via the `NSURLProtocolClient` protocol.

Subclasses must implement this method.

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
`- stopLoading` (page 13)

Declared In
`NSURLProtocol.h`

stopLoading

Stops protocol-specific loading of the request.

```
- (void)stopLoading
```

Discussion
When this method is called, the subclass implementation should stop loading a request. This could be in response to a cancel operation, so protocol implementations must be able to handle this call while a load is in progress.

Subclasses must implement this method.

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
`- startLoading` (page 13)

Declared In

`NSURLProtocol.h`

Document Revision History

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

Date	Notes
2007-04-01	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Index