WebPolicyDelegate Protocol Reference

Cocoa > User Experience

2009-04-08

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

WebPolicyDelegate Protocol Reference

(informal protocol)

Framework	/System/Library/Frameworks/WebKit.framework
Availability	Available in Mac OS X v10.2 with Safari 1.0 and later. Available in Mac OS X v10.2.7 and later.
Companion guide	WebKit Objective-C Programming Guide
Declared in	WebPolicyDelegate.h

Overview

The `WebPolicyDelegate` informal protocol works with the `WebPolicyDecisionListener` protocol to modify the policy decisions that the WebView class makes when handling URLs or the data objects they represent. The methods in this protocol are typically invoked in the following order.

1. The `webView:decidePolicyForNewWindowAction:request:newFrameName:decisionListener:` (page 8) method is invoked once for every load.
2. The `webView:decidePolicyForNavigationAction:request:frame:decisionListener:` (page 7) method may be invoked zero or more times after a load started. This method is invoked every time a server redirect is encountered unless blocked by an earlier policy decision.
3. The `webView:decidePolicyForMIMEType:request:frame:decisionListener:` (page 6) method is invoked after the MIME type of the content is known unless this method is blocked by an earlier policy decision.
4. The `webView:unableToImplementPolicyWithError:frame:` (page 8) method is invoked when an error occurs implementing a policy decision.

Tasks

Making Content Decisions

`- webView:decidePolicyForMIMEType:request:frame:decisionListener:` (page 6)
Decides whether to display content with a given MIME type.

Making Navigation Decisions

`- webView:decidePolicyForNavigationAction:request:frame:decisionListener:` (page 7)
Routes a navigation action internally or to an external viewer.

Making New Window Decisions

`- webView:decidePolicyForNewWindowAction:request:newFrameName:decisionListener:` (page 8)
Decides whether to allow a targeted navigation event, such as opening a link in a new window.

Handling Errors

`- webView:unableToImplementPolicyWithError:frame:` (page 8)
Handles or drops events that were rejected by a policy maker.

Instance Methods

webView:decidePolicyForMIMEType:request:frame:decisionListener:

Decides whether to display content with a given MIME type.

```
- (void)webView:(WebView *)webView decidePolicyForMIMEType:(NSString *)type
    request:(NSURLRequest *)request frame:(WebFrame *)frame decisionListener:(id
    < WebPolicyDecisionListener >)listener
```

Parameters

webView
The associated web view.

type
The MIME type of the content.

request
The request to load the content.

frame
The frame for displaying the content.

listener
The object that receives the policy decision.

Discussion
This method is invoked during the process of loading content for *request* after the `webView:didStartProvisionalLoadForFrame:` method in the `WebFrameLoadDelegate` informal protocol is called by the `WebView` object. The web view implements a policy decision by sending one of the `WebPolicyDecisionListener` protocol messages to *listener*.

If you do not implement this method, the default behavior is used. The listener is told to ignore the MIME type unless `webView` specifies it can handle the type in its `canShowMIMEType:` method.

In some rare cases, multiple responses may be received for a single resource. This happens in the case of multipart/x-mixed-replace, also known as a “server push.” In this case, this method will be invoked multiple times.

Availability
Available in Mac OS X v10.2 with Safari 1.0 and later.
Available in Mac OS X v10.2.7 and later.

Declared In
`WebPolicyDelegate.h`

webView:decidePolicyForNavigationAction:request:frame:decisionListener:

Routes a navigation action internally or to an external viewer.

```
- (void)webView:(WebView *)webView decidePolicyForNavigationAction:(NSDictionary
    *)actionInformation request:(NSURLRequest *)request frame:(WebFrame *)frame
    decisionListener:(id < WebPolicyDecisionListener >)listener
```

Parameters

webView
The `WebView` object for which this object is the policy delegate.

actionInformation
A description of the action that triggered the navigation request. The possible key-value pairs in this dictionary are defined in “Action Dictionary Keys” (page 9).

request
The request for which the navigation is made.

frame
The `WebFrame` object in which the action occurred.

listener
The `WebPolicyDecisionListener` object that receives the policy decision.

Discussion
This method is invoked when a navigation decision needs to be made. The web view implements a policy decision by sending one of the WebPolicyDecisionListener protocol messages to *listener*. This method is invoked whenever a server redirect is encountered, and before loading starts.

If you do not implement this method, the default behavior is used. The listener handles the navigation internally if the request is for an error page or if the `canHandleRequest:` method of the `NSURLConnection` class returns `YES` when passed `request`. Otherwise, the listener ignores the navigation, and it is handled externally.

Availability
Available in Mac OS X v10.2 with Safari 1.0 and later.
Available in Mac OS X v10.2.7 and later.

Declared In
`WebPolicyDelegate.h`

webView:decidePolicyForNewWindowAction:request:newFrameName:decisionListener:

Decides whether to allow a targeted navigation event, such as opening a link in a new window.

```
- (void)webView:(WebView *)webView decidePolicyForNewWindowAction:(NSDictionary
    *)actionInformation request:(NSURLRequest *)request newFrameName:(NSString
    *)frameName decisionListener:(id < WebPolicyDecisionListener >)listener
```

Parameters

webView

The `WebView` object for which this object is the policy delegate.

actionInformation

A description of the action that triggered the navigation request. The possible key-value pairs in this dictionary are defined in "Action Dictionary Keys" (page 9).

request

The request for which the new window action is performed.

frameName

The name of the new frame that contains the content returned from the request.

listener

The `WebPolicyDecisionListener` object that receives the policy decision.

Discussion

This method is invoked when a targeted navigation decision needs to be made. A targeted navigation typically opens a new window to display content. The receiver implements a policy decision by sending one of the WebPolicyDecisionListener protocol messages to *listener*. This method allows delegates to modify the behavior of targeted links which normally open a new window. Delegates might do something else, such as download or present the content in a special way. If this method sends `use` to *listener* then the new window will be opened, and `webView:decidePolicyForNavigationAction:request:frame:decisionListener:` (page 7) will be invoked with a `WebNavigationTypeOther` (page ?) as the value for the `WebActionNavigationTypeKey` (page ?) key in the action dictionary.

The default behavior sends `use` to *listener*.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

Declared In

`WebPolicyDelegate.h`

webView:unableToImplementPolicyWithError:frame:

Handles or drops events that were rejected by a policy maker.

```
- (void)webView:(WebView *)webView unableToImplementPolicyWithError:(NSError *)error
    frame:(WebFrame *)frame
```

Parameters

webView

The `WebView` object for which this object is the policy delegate.

error
The error that occurred.

frame
The frame in which the error occurred.

Discussion
Delegates might implement this method to display or log an error message. If you do not implement this method, no action is taken.

Availability
Available in Mac OS X v10.2 with Safari 1.0 and later.
Available in Mac OS X v10.2.7 and later.

Declared In
`WebPolicyDelegate.h`

Constants

Action Dictionary Keys

Keys that might appear in a dictionary passed as the `actionInformation` parameter to the `webView:decidePolicyForNavigationAction:request:frame:decisionListener:` (page 7) and `webView:decidePolicyForNewWindowAction:request:newFrameName:decisionListener:` (page 8) methods.

```
WebActionNavigationTypeKey
WebActionElementKey
WebActionButtonKey
WebActionModifierFlagsKey
WebActionOriginalURLKey
```

Constants
`WebActionNavigationTypeKey`
The navigation type of the action. Can be any of the values defined in "Navigation Type Values" below.
Available in Mac OS X v10.2 and later.
Declared in `WebPolicyDelegate.h`.

`WebActionElementKey`
A dictionary containing element information. See the "Constants" section of *WebView Class Reference* for a description of the key-value pairs in this dictionary.
Available in Mac OS X v10.2 and later.
Declared in `WebPolicyDelegate.h`.

`WebActionButtonKey`
The event type that occurred.
Available in Mac OS X v10.2 and later.
Declared in `WebPolicyDelegate.h`.

`WebActionModifierFlagsKey`
: An unsigned number that indicates the modifier flag.

  Available in Mac OS X v10.2 and later.

  Declared in `WebPolicyDelegate.h`.

`WebActionOriginalURLKey`
: The URL that initiated the action.

  Available in Mac OS X v10.2 and later.

  Declared in `WebPolicyDelegate.h`.

Navigation Type Values

These are the possible values for the `WebActionNavigationTypeKey` key that appears in an action dictionary.

```
WebNavigationTypeLinkClicked
WebNavigationTypeFormSubmitted
WebNavigationTypeBackForward
WebNavigationTypeReload
WebNavigationTypeFormResubmitted
WebNavigationTypeOther
```

Constants

`WebNavigationTypeLinkClicked`
: A link (an `href`) was clicked.

`WebNavigationTypeFormSubmitted`
: A form was submitted.

`WebNavigationTypeBackForward`
: The user clicked back or forward button.

`WebNavigationTypeReload`
: The user hit the reload button.

`WebNavigationTypeFormResubmitted`
: A form was resubmitted (through a back, forward or reload action).

`WebNavigationTypeOther`
: Navigation is taking place for some other reason.

Document Revision History

This table describes the changes to *WebPolicyDelegate Protocol Reference*.

Date	Notes
2009-04-08	Added separate parameter and return value descriptions.
2008-10-15	Minor edits throughout.
2006-05-23	First publication of this content as a separate document.

Document Revision History

Index