



Home  
Chromium  
Chromium OS

**Quick links**

Report bugs  
Discuss  
网站地图

**Other sites**

Chromium Blog  
Google Chrome  
Extensions

Except as otherwise [noted](#), the content of this page is licensed under a [Creative Commons Attribution 2.5 license](#), and examples are licensed under the [BSD License](#).

[For Developers](#) > [Design Documents](#) > [Desktop Notifications](#) >

## API Specification

**This is a DRAFT specification and subject to change.**

Note that the latest version of the *spec* is here: <http://dev.w3.org/2006/webapi/WebNotifications/publish/> but this page reflects what's implemented in the latest version of Chromium.

### The Notification interface

This interface represents a notification object.

```
interface Notification : EventTarget {
  // display methods
  void show();
  void cancel();

  // event handler attributes
  attribute Function ondisplay;
  attribute Function onerror;
  attribute Function onclose;
  attribute Function onclick;
}
```

#### Attributes

**ondisplay**

event listener function corresponding to event type "display". This listener must be called when the notification is displayed to the user, which need not be immediately when show() is called.

**onerror**

event listener function corresponding to event type "error". This listener must be called when the notification cannot be displayed to the user because of an error.

**onclose**

event listener function corresponding to event type "close". This listener must be called when the notification is closed by the user. This event must not occur until the "display" event.

**onclick**

event listener function corresponding to event type "click". This listener must be called when the notification has been clicked on by the user.

#### Methods

**show**

Causes the notification to displayed to the user. This may or may not happen immediately, depending on the constraints of the presentation method.

**cancel**

Causes the notification to not be displayed. If the notification has been displayed already, it must be closed. If it has not yet been displayed, it must be removed from the set of pending notifications.

### The NotificationCenter interface

The NotificationCenter interface exposes the ability for web pages to create Notification objects.

```
interface NotificationCenter {
  // Notification factory methods.
  Notification createNotification(in DOMString iconUrl, in DOMString
title, in DOMString body) throws(Exception);
  optional Notification createHTMLNotification(in DOMString url)
throws(Exception);
}
```

```
// Permission values
const unsigned int PERMISSION_ALLOWED = 0;
const unsigned int PERMISSION_NOT_ALLOWED = 1;
const unsigned int PERMISSION_DENIED = 2;

// Permission methods
int checkPermission();
void requestPermission(in Function callback);
}
```

The NotificationCenter interface is available via the Window interface through the `webkitNotifications` property.

```
interface Window {
...
  attribute NotificationCenter webkitNotifications;
...
}
```

#### Methods

##### **createNotification**

Creates a new notification object with the provided content. *iconUrl* contains the URL of an image resource to be shown with the notification; *title* contains a string which is the primary text of the notification; *body* contains a string which is secondary text for the notification. If the origin of the script which executes this method does not have permission level `PERMISSION_ALLOWED`, this method will throw a security exception.

##### **createHTMLNotification**

The parameter *url* contains the URL of a resource which contains HTML content to be shown as the content of the notification. If the origin of the script which executes this method does not have permission level `PERMISSION_ALLOWED`, this method will throw a security exception. In Chromium, only `http:`, `https:`, and `chrome-extension:` schemes are allowed for *url*.

##### **checkPermission**

Returns an integer, either `PERMISSION_ALLOWED`, `PERMISSION_NOT_ALLOWED`, or `PERMISSION_DENIED`, which indicates the permission level granted to the origin of the script currently executing.

- `PERMISSION_ALLOWED` (0) indicates that the user has granted permission to scripts with this origin to show notifications.
- `PERMISSION_NOT_ALLOWED` (1) indicates that the user has not taken an action regarding notifications for scripts from this origin.
- `PERMISSION_DENIED` (2) indicates that the user has explicitly blocked scripts with this origin from showing notifications.

##### **requestPermission**

Requests that the user agent ask the user for permission to show notifications from scripts. This method should only be called while handling a user gesture; in other circumstances it will have no effect. This method is asynchronous. The function provided in *callback* will be invoked when the user has responded to the permission request. If the current permission level is `PERMISSION_DENIED`, the user agent may take no action in response to `requestPermission`.

## Lifetime of notifications

If the page which generates a notification was generated is closed, the notification itself will not be closed, but events will not be delivered, as the script execution context is no longer present.

## 评论

[登录](#) | [最近的网站活动](#) | [举报滥用行为](#) | [打印页面](#) | 由 [Google 协作平台](#) 强力驱动