

<< [API](#)

<< **Web Page Module**

- P

[canGoBack](#)
- P

[canGoForward](#)
- P

[clipRect](#)
- P

[content](#)
- P

[cookies](#)
- P

[customHeaders](#)
- P

[event](#)
- P

[focusedFrameName](#)
- P

[frameContent](#)
- P

[frameName](#)
- P

[framePlainText](#)
- P

[frameTitle](#)
- P

[frameUrl](#)
- P

[framesCount](#)
- P

[framesName](#)
- P

[libraryPath](#)
- P

[navigationLocked](#)
- P

[offlineStoragePath](#)
- P

[offlineStorageQuota](#)
- P

[ownsPages](#)
- P

[pagesWindowName](#)
- P

[pages](#)
- P

[paperSize](#)
- P

[plainText](#)
- P

[scrollPosition](#)
- P

[settings](#)
- P

[title](#)
- P

[url](#)
- P

[viewportSize](#)
- P

[windowName](#)
- P

[zoomFactor](#)
- M

[addCookie](#)
- M

[childFramesCount](#)
- M

[childFramesName](#)
- M

[clearCookies](#)
- M

[close](#)
- M

[currentFrameName](#)
- M

[deleteCookie](#)
- M

[evaluateAsync](#)
- M

[evaluateJavaScript](#)
- M

[evaluate](#)
- M

[getPage](#)
- M

[goBack](#)
- M

[goForward](#)
- M

[go](#)
- M

[includeJs](#)
- M

[injectJs](#)
- M

[openUrl](#)
- M

[open](#)
- M

[release](#)
- M

[reload](#)
- M

[renderBase64](#)
- M

[renderBuffer](#)
- M

[render](#)
- M

[sendEvent](#)
- M

[setContent](#)
- M

[stop](#)
- M

[switchToChildFrame](#)
- M

[switchToFocusedFrame](#)
- M

[switchToFrame](#)
- M

[switchToMainFrame](#)
- M

[switchToParentFrame](#)
- M

[uploadFile](#)
- H

[Callback Triggers](#)
- H

[onAlert](#)
- H

[onCallback](#)
- H

[onClosing](#)
- H

[onConfirm](#)
- H

[onConsoleMessage](#)
- H

[onError](#)
- H

[onFilePicker](#)
- H

[onInitialized](#)
- H

[onLoadFinished](#)
- H

[onLoadStarted](#)
- H

[onNavigationRequested](#)
- H

[onPageCreated](#)
- H

[onPrompt](#)
- H

[onResourceError](#)
- H

[onResourceReceived](#)
- H

[onResourceRequested](#)
- H

[onResourceTimeout](#)
- H

[onUrlChanged](#)

# onLoadFinished

**Introduced:** PhantomJS 1.2

This callback is invoked when the page finishes the loading. It may accept a single argument indicating the page's `status`: `'success'` if no network errors occurred, otherwise `'fail'`.

Also see `page.open` for an alternate hook for the `onLoadFinished` callback.

## Examples

```
var webPage = require('webpage');
var page = webPage.create();

page.onLoadFinished = function(status) {
    console.log('Status: ' + status);
    // Do other things here...
};
```