

Class: MenuItem

Add items to native application menus and context menus.

Process: [Main](#)

See [Menu](#) for examples.

new MenuItem(options)

- options Object
 - click Function (optional) - Will be called with `click(menuItem, browserWindow, event)` when the menu item is clicked.
 - menuItem MenuItem
 - browserWindow [BrowserWindow](#) | undefined - This will not be defined if no window is open.
 - event [KeyboardEvent](#)
 - role string (optional) - Can be `undo`, `redo`, `cut`, `copy`, `paste`, `pasteAndMatchStyle`, `delete`, `selectAll`, `reload`, `forceReload`, `toggleDevTools`, `resetZoom`, `zoomIn`, `zoomOut`, `toggleSpellChecker`, `togglefullscreen`, `window`, `minimize`, `close`, `help`, `about`, `services`, `hide`, `hideOthers`, `unhide`, `quit`, `'showSubstitutions'`, `'toggleSmartQuotes'`, `'toggleSmartDashes'`, `'toggleTextReplacement'`, `startSpeaking`, `stopSpeaking`, `zoom`, `front`, `appMenu`, `fileMenu`, `editMenu`, `viewMenu`, `shareMenu`, `recentDocuments`, `toggleTabBar`, `selectNextTab`, `selectPreviousTab`, `mergeAllWindows`, `clearRecentDocuments`, `moveTabToNewWindow` or `windowMenu` - Define the action of the menu item, when specified the `click` property will be ignored. See [roles](#).
 - type string (optional) - Can be `normal`, `separator`, `submenu`, `checkbox` or `radio`.
 - label string (optional)
 - sublabel string (optional)
 - toolTip string (optional) *macOS* - Hover text for this menu item.
 - accelerator [Accelerator](#) (optional)
 - icon ([NativeImage](#) | string) (optional)
 - enabled boolean (optional) - If false, the menu item will be greyed out and unclickable.
 - acceleratorWorksWhenHidden boolean (optional) *macOS* - default is `true`, and when `false` will prevent the accelerator from triggering the item if the item is not visible.
 - visible boolean (optional) - If false, the menu item will be entirely hidden.
 - checked boolean (optional) - Should only be specified for `checkbox` or `radio` type menu items.
 - registerAccelerator boolean (optional) *Linux Windows* - If false, the accelerator won't be registered with the system, but it will still be displayed. Defaults to true.
 - sharingItem SharingItem (optional) *macOS* - The item to share when the `role` is `shareMenu`.
 - submenu (MenuItemConstructorOptions[] | [Menu](#)) (optional) - Should be specified for `submenu` type menu items. If `submenu` is specified, the `type: 'submenu'` can be omitted. If the value is not a [Menu](#) then it will be automatically converted to one using `Menu.buildFromTemplate`.

- `id` string (optional) - Unique within a single menu. If defined then it can be used as a reference to this item by the position attribute.
- `before` string[] (optional) - Inserts this item before the item with the specified label. If the referenced item doesn't exist the item will be inserted at the end of the menu. Also implies that the menu item in question should be placed in the same "group" as the item.
- `after` string[] (optional) - Inserts this item after the item with the specified label. If the referenced item doesn't exist the item will be inserted at the end of the menu.
- `beforeGroupContaining` string[] (optional) - Provides a means for a single context menu to declare the placement of their containing group before the containing group of the item with the specified label.
- `afterGroupContaining` string[] (optional) - Provides a means for a single context menu to declare the placement of their containing group after the containing group of the item with the specified label.

Note: `acceleratorWorksWhenHidden` is specified as being macOS-only because accelerators always work when items are hidden on Windows and Linux. The option is exposed to users to give them the option to turn it off, as this is possible in native macOS development. This property is only usable on macOS High Sierra 10.13 or newer.

Roles

Roles allow menu items to have predefined behaviors.

It is best to specify `role` for any menu item that matches a standard role, rather than trying to manually implement the behavior in a `click` function. The built-in `role` behavior will give the best native experience.

The `label` and `accelerator` values are optional when using a `role` and will default to appropriate values for each platform.

Every menu item must have either a `role`, `label`, or in the case of a separator a `type`.

The `role` property can have following values:

- `undo`
- `about` - Trigger a native about panel (custom message box on Window, which does not provide its own).
- `redo`
- `cut`
- `copy`
- `paste`
- `pasteAndMatchStyle`
- `selectAll`
- `delete`
- `minimize` - Minimize current window.
- `close` - Close current window.
- `quit` - Quit the application.
- `reload` - Reload the current window.
- `forceReload` - Reload the current window ignoring the cache.
- `toggleDevTools` - Toggle developer tools in the current window.
- `togglefullscreen` - Toggle full screen mode on the current window.
- `resetZoom` - Reset the focused page's zoom level to the original size.
- `zoomIn` - Zoom in the focused page by 10%.

- `zoomOut` - Zoom out the focused page by 10%.
- `toggleSpellChecker` - Enable/disable builtin spell checker.
- `fileMenu` - Whole default "File" menu (Close / Quit)
- `editMenu` - Whole default "Edit" menu (Undo, Copy, etc.).
- `viewMenu` - Whole default "View" menu (Reload, Toggle Developer Tools, etc.)
- `windowMenu` - Whole default "Window" menu (Minimize, Zoom, etc.).

The following additional roles are available on *macOS*:

- `appMenu` - Whole default "App" menu (About, Services, etc.)
- `hide` - Map to the `hide` action.
- `hideOthers` - Map to the `hideOtherApplications` action.
- `unhide` - Map to the `unhideAllApplications` action.
- `showSubstitutions` - Map to the `orderFrontSubstitutionsPanel` action.
- `toggleSmartQuotes` - Map to the `toggleAutomaticQuoteSubstitution` action.
- `toggleSmartDashes` - Map to the `toggleAutomaticDashSubstitution` action.
- `toggleTextReplacement` - Map to the `toggleAutomaticTextReplacement` action.
- `startSpeaking` - Map to the `startSpeaking` action.
- `stopSpeaking` - Map to the `stopSpeaking` action.
- `front` - Map to the `arrangeInFront` action.
- `zoom` - Map to the `performZoom` action.
- `toggleTabBar` - Map to the `toggleTabBar` action.
- `selectNextTab` - Map to the `selectNextTab` action.
- `selectPreviousTab` - Map to the `selectPreviousTab` action.
- `mergeAllWindows` - Map to the `mergeAllWindows` action.
- `moveTabToNewWindow` - Map to the `moveTabToNewWindow` action.
- `window` - The submenu is a "Window" menu.
- `help` - The submenu is a "Help" menu.
- `services` - The submenu is a "[Services](#)" menu. This is only intended for use in the Application Menu and is *not* the same as the "Services" submenu used in context menus in macOS apps, which is not implemented in Electron.
- `recentDocuments` - The submenu is an "Open Recent" menu.
- `clearRecentDocuments` - Map to the `clearRecentDocuments` action.
- `shareMenu` - The submenu is [share menu](#). The `sharingItem` property must also be set to indicate the item to share.

When specifying a `role` on macOS, `label` and `accelerator` are the only options that will affect the menu item. All other options will be ignored. Lowercase `role`, e.g. `toggleddevtools`, is still supported.

Note: The `enabled` and `visibility` properties are not available for top-level menu items in the tray on macOS.

Instance Properties

The following properties are available on instances of `MenuItem` :

`menuItem.id`

A `string` indicating the item's unique id, this property can be dynamically changed.

`menuItem.label`

A `string` indicating the item's visible label.

`menuItem.click`

A `Function` that is fired when the MenuItem receives a click event. It can be called with

`menuItem.click(event, focusedWindow, focusedWebContents)` .

- `event` [KeyboardEvent](#)
- `focusedWindow` [BrowserWindow](#)
- `focusedWebContents` [WebContents](#)

`menuItem.submenu`

A `Menu` (optional) containing the menu item's submenu, if present.

`menuItem.type`

A `string` indicating the type of the item. Can be `normal` , `separator` , `submenu` , `checkbox` or `radio` .

`menuItem.role`

A `string` (optional) indicating the item's role, if set. Can be `undo` , `redo` , `cut` , `copy` , `paste` , `pasteAndMatchStyle` , `delete` , `selectAll` , `reload` , `forceReload` , `toggleDevTools` , `resetZoom` , `zoomIn` , `zoomOut` , `toggleSpellChecker` , `toggleFullscreen` , `window` , `minimize` , `close` , `help` , `about` , `services` , `hide` , `hideOthers` , `unhide` , `quit` , `startSpeaking` , `stopSpeaking` , `zoom` , `front` , `appMenu` , `fileMenu` , `editMenu` , `viewMenu` , `shareMenu` , `recentDocuments` , `toggleTabBar` , `selectNextTab` , `selectPreviousTab` , `mergeAllWindows` , `clearRecentDocuments` , `moveTabToNewWindow` or `windowMenu`

`menuItem.accelerator`

An `Accelerator` (optional) indicating the item's accelerator, if set.

`menuItem.userAccelerator` **Readonly macOS**

An `Accelerator` | `null` indicating the item's [user-assigned accelerator](#) for the menu item.

Note: This property is only initialized after the `MenuItem` has been added to a `Menu` . Either via `Menu.buildFromTemplate` or via `Menu.append()`/`insert()` . Accessing before initialization will just return `null` .

`menuItem.icon`

A `NativeImage` | `string` (optional) indicating the item's icon, if set.

`menuItem.sublabel`

A `string` indicating the item's sublabel.

`menuItem.toolTip` **macOS**

A `string` indicating the item's hover text.

`menuItem.enabled`

A `boolean` indicating whether the item is enabled, this property can be dynamically changed.

`menuItem.visible`

A `boolean` indicating whether the item is visible, this property can be dynamically changed.

`menuItem.checked`

A `boolean` indicating whether the item is checked, this property can be dynamically changed.

A `checkbox` menu item will toggle the `checked` property on and off when selected.

A `radio` menu item will turn on its `checked` property when clicked, and will turn off that property for all adjacent items in the same menu.

You can add a `click` function for additional behavior.

`menuItem.registerAccelerator`

A `boolean` indicating if the accelerator should be registered with the system or just displayed.

This property can be dynamically changed.

`menuItem.sharingItem` *macOS*

A `SharingItem` indicating the item to share when the `role` is `shareMenu`.

This property can be dynamically changed.

`menuItem.commandId`

A `number` indicating an item's sequential unique id.

`menuItem.menu`

A `Menu` that the item is a part of.