Chrome Extension Support

Electron supports a subset of the Chrome Extensions API, primarily to support DevTools extensions and Chromium-internal extensions, but it also happens to support some other extension capabilities.

> ***Note:*** *Electron does not support arbitrary Chrome extensions from the store, and it is a* ***non-goal*** *of the Electron project to be perfectly compatible with Chrome's implementation of Extensions.*

Loading extensions

Electron only supports loading unpacked extensions (i.e., `.crx` files do not work). Extensions are installed per-`session`. To load an extension, call `ses.loadExtension`:

```
const { session } = require('electron')

session.loadExtension('path/to/unpacked/extension').then(({ id }) => {
  // ...
})
```

Loaded extensions will not be automatically remembered across exits; if you do not call `loadExtension` when the app runs, the extension will not be loaded.

Note that loading extensions is only supported in persistent sessions. Attempting to load an extension into an in-memory session will throw an error.

See the `session` documentation for more information about loading, unloading, and querying active extensions.

Supported Extensions APIs

We support the following extensions APIs, with some caveats. Other APIs may additionally be supported, but support for any APIs not listed here is provisional and may be removed.

`chrome.devtools.inspectedWindow`

All features of this API are supported.

`chrome.devtools.network`

All features of this API are supported.

`chrome.devtools.panels`

All features of this API are supported.

`chrome.extension`

The following properties of `chrome.extension` are supported:

- `chrome.extension.lastError`

The following methods of `chrome.extension` are supported:

- `chrome.extension.getURL`

- `chrome.extension.getBackgroundPage`

`chrome.runtime`

The following properties of `chrome.runtime` are supported:

- `chrome.runtime.lastError`
- `chrome.runtime.id`

The following methods of `chrome.runtime` are supported:

- `chrome.runtime.getBackgroundPage`
- `chrome.runtime.getManifest`
- `chrome.runtime.getPlatformInfo`
- `chrome.runtime.getURL`
- `chrome.runtime.connect`
- `chrome.runtime.sendMessage`
- `chrome.runtime.reload`

The following events of `chrome.runtime` are supported:

- `chrome.runtime.onStartup`
- `chrome.runtime.onInstalled`
- `chrome.runtime.onSuspend`
- `chrome.runtime.onSuspendCanceled`
- `chrome.runtime.onConnect`
- `chrome.runtime.onMessage`

`chrome.storage`

Only `chrome.storage.local` is supported; `chrome.storage.sync` and `chrome.storage.managed` are not.

`chrome.tabs`

The following methods of `chrome.tabs` are supported:

- `chrome.tabs.sendMessage`
- `chrome.tabs.executeScript`
- `chrome.tabs.update` (partial support)
  - supported properties: `url`, `muted`.

> ***Note:*** *In Chrome, passing `-1` as a tab ID signifies the "currently active tab". Since Electron has no such concept, passing `-1` as a tab ID is not supported and will raise an error.*

`chrome.management`

The following methods of `chrome.management` are supported:

- `chrome.management.getAll`
- `chrome.management.get`
- `chrome.management.getSelf`
- `chrome.management.getPermissionWarningsById`

- `chrome.management.getPermissionWarningsByManifest`
- `chrome.management.onEnabled`
- `chrome.management.onDisabled`

`chrome.webRequest`

All features of this API are supported.

> ***NOTE:*** *Electron's* `webRequest` *module takes precedence over* `chrome.webRequest` *if there are conflicting handlers.*