autoUpdater

Enable apps to automatically update themselves.

Process: Main

See also: A detailed guide about how to implement updates in your application.

`autoUpdater` is an EventEmitter.

Platform Notices

Currently, only macOS and Windows are supported. There is no built-in support for auto-updater on Linux, so it is recommended to use the distribution's package manager to update your app.

In addition, there are some subtle differences on each platform:

macOS

On macOS, the `autoUpdater` module is built upon Squirrel.Mac, meaning you don't need any special setup to make it work. For server-side requirements, you can read Server Support. Note that App Transport Security (ATS) applies to all requests made as part of the update process. Apps that need to disable ATS can add the `NSAllowsArbitraryLoads` key to their app's plist.

Note: Your application must be signed for automatic updates on macOS. This is a requirement of `Squirrel.Mac`.

Windows

On Windows, you have to install your app into a user's machine before you can use the `autoUpdater`, so it is recommended that you use the electron-winstaller, electron-forge or the grunt-electron-installer package to generate a Windows installer.

When using electron-winstaller or electron-forge make sure you do not try to update your app the first time it runs (Also see this issue for more info). It's also recommended to use electron-squirrel-startup to get desktop shortcuts for your app.

The installer generated with Squirrel will create a shortcut icon with an Application User Model ID in the format of `com.squirrel.PACKAGE_ID.YOUR_EXE_WITHOUT_DOT_EXE`, examples are `com.squirrel.slack.Slack` and `com.squirrel.code.Code`. You have to use the same ID for your app with `app.setAppUserModelId` API, otherwise Windows will not be able to pin your app properly in task bar.

Like Squirrel.Mac, Windows can host updates on S3 or any other static file host. You can read the documents of Squirrel.Windows to get more details about how Squirrel.Windows works.

Events

The `autoUpdater` object emits the following events:

Event: 'error'

Returns:

- `error` Error

Emitted when there is an error while updating.

Event: 'checking-for-update'

Emitted when checking if an update has started.

Event: 'update-available'

Emitted when there is an available update. The update is downloaded automatically.

Event: 'update-not-available'

Emitted when there is no available update.

Event: 'update-downloaded'

Returns:

- `event` Event
- `releaseNotes` string
- `releaseName` string
- `releaseDate` Date
- `updateURL` string

Emitted when an update has been downloaded.

On Windows only `releaseName` is available.

Note: It is not strictly necessary to handle this event. A successfully downloaded update will still be applied the next time the application starts.

Event: 'before-quit-for-update'

This event is emitted after a user calls `quitAndInstall()`.

When this API is called, the `before-quit` event is not emitted before all windows are closed. As a result you should listen to this event if you wish to perform actions before the windows are closed while a process is quitting, as well as listening to `before-quit`.

Methods

The `autoUpdater` object has the following methods:

`autoUpdater.setFeedURL(options)`

- `options` Object
  - `url` string
  - `headers` Record<string, string> (optional) *macOS* - HTTP request headers.
  - `serverType` string (optional) *macOS* - Can be `json` or `default`, see the Squirrel.Mac README for more information.

Sets the `url` and initialize the auto updater.

`autoUpdater.getFeedURL()`

Returns `string` - The current update feed URL.

`autoUpdater.checkForUpdates()`

Asks the server whether there is an update. You must call `setFeedURL` before using this API.

Note: If an update is available it will be downloaded automatically. Calling `autoUpdater.checkForUpdates()` twice will download the update two times.

`autoUpdater.quitAndInstall()`

Restarts the app and installs the update after it has been downloaded. It should only be called after `update-downloaded` has been emitted.

Under the hood calling `autoUpdater.quitAndInstall()` will close all application windows first, and automatically call `app.quit()` after all windows have been closed.

Note: It is not strictly necessary to call this function to apply an update, as a successfully downloaded update will always be applied the next time the application starts.