

Configuration

The Tauri configuration object. It is read from a file where you can define your frontend assets, configure the bundler, enable the app updater, define a system tray, enable APIs via the allowlist and more.

The configuration file is generated by the `tauri init` command that lives in your Tauri application source directory (src-tauri).

Once generated, you may modify it at will to customize your Tauri application.

File Formats

By default, the configuration is defined as a JSON file named `tauri.conf.json`.

Tauri also supports JSON5 and TOML files via the `config-json5` and `config-toml` Cargo features, respectively. The JSON5 file name must be either `tauri.conf.json` or `tauri.conf.json5`. The TOML file name is `Tauri.toml`.

Platform-Specific Configuration

In addition to the default configuration file, Tauri can read a platform-specific configuration from `tauri.linux.conf.json`, `tauri.windows.conf.json`, and `tauri.macos.conf.json` (or `Tauri.linux.toml`, `Tauri.windows.toml` and `Tauri.macos.toml` if the `Tauri.toml` format is used), which gets merged with the main configuration object.

Configuration Structure

The configuration is composed of the following objects:

- `package`: Package settings
- `tauri`: The Tauri config
- `build`: The build configuration
- `plugins`: The plugins config

Example `tauri.config.json` file

```
{
  "build": {
    "beforeBuildCommand": "",
    "beforeDevCommand": "",
    "devPath": "../dist",
    "distDir": "../dist"
  },
  "package": {
    "productName": "tauri-app",
    "version": "0.1.0"
  },
  "tauri": {
    "allowlist": {
      "all": true
    },
    "bundle": {},
    "security": {
      "csp": null
    },
    "updater": {
      "active": false
    },
    "windows": [
      {
        "fullscreen": false,
        "height": 600,
        "resizable": true,
        "title": "Tauri App",
        "width": 800
      }
    ]
  }
}
```

Type: object

Name	Type	Default	Description
package	PackageConfig	view	Package settings.
tauri	TauriConfig	view	The Tauri configuration.
build	BuildConfig	view	The build configuration.
plugins	PluginConfig	view	The plugins config.

PackageConfig

The package configuration.

Type: object

Name	Type	Default	Description
productName	string?	null	App name.
version	string?	null	App version. It is a semver version number or a path to a <code>package.json</code> file containing the <code>version</code> field. If removed the version number from <code>Cargo.toml</code> is used.

TauriConfig

The Tauri configuration object.

Type: object

Name	Type	Default	Description
pattern	PatternKind	view	The pattern to use.
windows	WindowConfig	[]	The windows configuration.
cli	CliConfig?	view	The CLI configuration.
bundle	BundleConfig	view	The bundler configuration.
allowlist	AllowlistConfig	view	The allowlist configuration.
security	SecurityConfig	view	Security configuration.
updater	UpdaterConfig	view	The updater configuration.
systemTray	SystemTrayConfig?	view	Configuration for app system tray.
macOSPrivateApi	boolean	false	MacOS private API configuration. Enables the transparent background API and sets the

Name	Type	Default	Description
			fullScreenEnabled preference to true.

PatternKind

The application pattern.

Can be any **ONE** of the following types:

- `{ "use": "brownfield" }`: Brownfield pattern.

Name	Type	Default	Description
<code>use</code>	"brownfield" (required)		undefined

- `{ "use": "isolation", "options": { "dir": string } }`: Isolation pattern. Recommended for security purposes.

Name	Type	Default	Description
<code>use</code>	"isolation" (required)		undefined
<code>options</code>	<code>{ "dir": string }</code> (required)		undefined

WindowConfig

The window configuration object.

Type: `object`

Name	Type	Default	Description
<code>label</code>	<code>string</code>	<code>null</code>	The window identifier. It must be alphanumeric.
<code>url</code>	<code>WindowUrl</code>	<code>view</code>	The window webview URL.
<code>userAgent</code>	<code>string?</code>	<code>null</code>	The user agent for the webview

Name	Type	Default	Description
fileDropEnabled	boolean	true	Whether the file drop is enabled or not on the windows. By default it is enabled.
			Disabling it is required to use drag and drop on frontend on Windows.
center	boolean	false	Whether or not the window starts centered or not.
x	number? (format: double)	null	The horizontal position of the window's top left corner.
y	number? (format: double)	null	The vertical position of the window's top left corner.
width	number (format: double)	800	The window width.
height	number (format: double)	600	The window height.
minWidth	number? (format: double)	null	The min window width.
minHeight	number? (format: double)	null	The min window height.
maxWidth	number? (format: double)	null	The max window width.
maxHeight	number? (format: double)	null	The max window height.

Name	Type	Default	Description
	<code>double</code>		
<code>resizable</code>	boolean	<code>true</code>	Whether the window is resizable or not. When <code>resizable</code> is set to false, native window's maximize button will automatically disabled.
			Whether the window's native maximize button is enabled or not. If <code>resizable</code> is set to false, this setting is ignored.
<code>maximizable</code>	boolean	<code>true</code>	<p>## Platform-specific</p> <ul style="list-style-type: none"> - macOS: Disables the "zoom" button in the window titlebar, which is also used to enter fullscreen mode. - Linux / iOS / Android: Unsupported.
			Whether the window's native minimize button is enabled or not.
<code>minimizable</code>	boolean	<code>true</code>	<p>## Platform-specific</p> <ul style="list-style-type: none"> - Linux / iOS / Android: Unsupported.
			Whether the window's native close button is enabled or not.
<code>closable</code>	boolean	<code>true</code>	<p>## Platform-specific</p> <ul style="list-style-type: none"> - Linux: "GTK+ will do its best to convince the window manager not to show a close button. Depending on the system, this function may not have any effect if called on a window that is already visible" - iOS / Android: Unsupported.
<code>title</code>	string	<code>null</code>	The window title.
<code>fullscreen</code>	boolean	<code>false</code>	Whether the window starts as fullscreen or not.

Name	Type	Default	Description
focus	boolean	true	Whether the window will be initially focused or not.
transparent	boolean	false	Whether the window is transparent or not. Note that on macOS this requires the <code>macos-private-api</code> feature flag, enabled under <code>tauri > macOSPrivateAPI</code> . WARNING: Using private APIs on macOS prevents your application from being accepted to the App Store.
maximized	boolean	false	Whether the window is maximized or not.
visible	boolean	true	Whether the window is visible or not.
decorations	boolean	true	Whether the window should have borders and decorations.
alwaysOnTop	boolean	false	Whether the window should always be on top of other windows.
contentProtected	boolean	false	Prevents the window contents from being captured by other apps.
skipTaskbar	boolean	false	If true, hides the window icon from the taskbar on Windows and Linux.
theme	Theme?	view	The initial window theme. Defaults to the system theme. Only implemented on Windows and macOS 10.14+.
titleBarStyle	TitleBarStyle	view	The style of the macOS title bar.
hiddenTitle	boolean	false	If true, sets the window title to be hidden on macOS.
acceptFirstMouse	boolean	false	Whether clicking an inactive window also clicks to the webview on macOS.
tabbingIdentifier	string?	null	Defines the window tabbing identifier for macOS. Windows with matching tabbing identifiers will...

Name	Type	Default	Description
			grouped together. If the tabbing identifier is no automatic tabbing will be disabled.
additionalBrowserArgs	string?	null	Defines additional browser arguments on Windows default wry passes <code>--disable-features=msWebOOUI,msPdfOOUI,msSmartScreenPlugin</code> so if you use this method, you also need to disable these components by yourself if you want.

WindowUrl

An URL to open on a Tauri webview window.

Can be any of the following types:

- `string (format: uri)`: An external URL.
- `string`: The path portion of an app URL. For instance, to load `tauri://localhost/users/john`, you can simply provide `users/john` in this configuration.

Theme

System theme.

Can be any **ONE** of the following types:

- "Light": Light theme.
- "Dark": Dark theme.

TitleBarStyle

How the window title bar should be displayed on macOS.

Can be any **ONE** of the following types:

- "Visible": A normal title bar.
- "Transparent": Makes the title bar transparent, so the window background color is shown instead.

Useful if you don't need to have actual HTML under the title bar. This lets you avoid the caveats of using `TitleBarStyle::Overlay`. Will be more useful when Tauri lets you set a custom window

background color.

- "Overlay": Shows the title bar as a transparent overlay over the window's content.

Keep in mind:

- The height of the title bar is different on different OS versions, which can lead to window controls and title not being where you don't expect.
- You need to define a custom drag region to make your window draggable, however due to a limitation you can't drag the window when it's not in focus <https://github.com/tauri-apps/tauri/issues/4316>.
- The color of the window title depends on the system theme.

CliConfig

describes a CLI configuration

Type: object

Name	Type	Default	Description
description	string?	null	Command description which will be shown on the help information.
longDescription	string?	null	Command long description which will be shown on the help information.
beforeHelp	string?	null	Adds additional help information to be displayed in addition to auto-generated help. This information is displayed before the auto-generated help information. This is often used for header information.
afterHelp	string?	null	Adds additional help information to be displayed in addition to auto-generated help. This information is displayed after the auto-generated help information. This is often used to describe how to use the arguments, or caveats to be noted.
args	CliArg?	null	List of arguments for the command

Name	Type	Default	Description
subcommands	CliConfig?	null	List of subcommands of this command

CliArg

A CLI argument definition.

Type: object

Name	Type	Default	Description
short	string?	null	<p>The short version of the argument, without the preceding -.</p> <p>NOTE: Any leading - characters will be stripped, and only the first non-character will be used as the short version.</p>
name	string (required)		The unique argument name
description	string?	null	The argument description which will be shown on the help information. Typically, this is a short (one line) description of the arg.
longDescription	string?	null	The argument long description which will be shown on the help information. Typically this a more detailed (multi-line) message that describes the argument.
takesValue	boolean	false	<p>Specifies that the argument takes a value at run time.</p> <p>NOTE: values for arguments may be specified in any of the following methods</p> <ul style="list-style-type: none"> - Using a space such as -o value or --option value

Name	Type	Default	Description
			<ul style="list-style-type: none"> - Using an equals and no space such as -o=value or --option=value - Use a short and no space such as -ovalue
<code>multiple</code>	<code>boolean</code>	<code>false</code>	<p>Specifies that the argument may have an unknown number of multiple values. Without any other settings, this argument may appear only once.</p> <p>For example, <code>--opt val1 val2</code> is allowed, but <code>--opt val1 val2 --opt val3</code> is not.</p> <p>NOTE: Setting this requires <code>takes_value</code> to be set to true.</p>
<code>multipleOccurrences</code>	<code>boolean</code>	<code>false</code>	<p>Specifies that the argument may appear more than once. For flags, this results in the number of occurrences of the flag being recorded. For example <code>-ddd</code> or <code>-d -d -d</code> would count as three occurrences. For options or arguments that take a value, this does not affect how many values they can accept. (i.e. only one at a time is allowed)</p> <p>For example, <code>--opt val1 --opt val2</code> is allowed, but <code>--opt val1 val2</code> is not.</p>
<code>numberOfValues</code>	<code>integer?</code> <i>(format: uint)</i>	<code>null</code>	<p>Specifies how many values are required to satisfy this argument. For example, if you had a <code>-f <file></code> argument where you wanted exactly 3 'files' you would set <code>number_of_values = 3</code>, and this argument wouldn't be satisfied unless the user provided 3 and only 3 values.</p> <p>NOTE: Does <i>not</i> require <code>multiple_occurrences = true</code> to be set. Setting <code>multiple_occurrences = true</code> would</p>

Name	Type	Default	Description
			<p>allow <code>-f <file> <file> <file> -f <file> <file> <file></code> where as <i>not</i> setting it would only allow one occurrence of this argument.</p> <p>NOTE: implicitly sets <code>takes_value = true</code> and <code>multiple_values = true</code>.</p>
<code>possibleValues</code>	array?	<code>null</code>	Specifies a list of possible values for this argument. At runtime, the CLI verifies that only one of the specified values was used, or fails with an error message.
<code>minValues</code>	integer? <i>(format: uint)</i>	<code>null</code>	Specifies the minimum number of values for this argument. For example, if you had a <code>-f <file></code> argument where you wanted at least 2 'files', you would set <code>minValues: 2</code> , and this argument would be satisfied if the user provided, 2 or more values.
<code>maxValues</code>	integer? <i>(format: uint)</i>	<code>null</code>	Specifies the maximum number of values are for this argument. For example, if you had a <code>-f <file></code> argument where you wanted up to 3 'files', you would set <code>.max_values(3)</code> , and this argument would be satisfied if the user provided, 1, 2, or 3 values.
<code>required</code>	boolean	<code>false</code>	<p>Sets whether or not the argument is required by default.</p> <ul style="list-style-type: none"> - Required by default means it is required, when no other conflicting rules have been evaluated - Conflicting rules take precedence over being required.
<code>requiredUnlessPresent</code>	string?	<code>null</code>	Sets an arg that override this arg's required setting i.e. this arg will be required unless

Name	Type	Default	Description
			this other argument is present.
requiredUnlessPresentAll	array?	null	Sets args that override this arg's required setting i.e. this arg will be required unless all these other arguments are present.
requiredUnlessPresentAny	array?	null	Sets args that override this arg's required setting i.e. this arg will be required unless at least one of these other arguments are present.
conflictsWith	string?	null	Sets a conflicting argument by name i.e. when using this argument, the following argument can't be present and vice versa.
conflictsWithAll	array?	null	The same as conflictsWith but allows specifying multiple two-way conflicts per argument.
requires	string?	null	Tets an argument by name that is required when this one is present i.e. when using this argument, the following argument must be present.
requiresAll	array?	null	Sts multiple arguments by names that are required when this one is present i.e. when using this argument, the following arguments must be present.
requiresIf	array?	null	Allows a conditional requirement with the signature [arg, value] the requirement will only become valid if arg's value equals \${value} .
requiredIfEq	array?	null	Allows specifying that an argument is required conditionally with the signature [arg, value] the requirement will only

Name	Type	Default	Description
			become valid if the <code>arg</code> 's value equals <code> \${value} </code> .
<code>requireEquals</code>	<code>boolean?</code>	<code>null</code>	Requires that options use the <code>--option=val</code> syntax i.e. an equals between the option and associated value.
<code>index</code>	<code>integer?</code> <i>(format:</i> <code>uint</code> , <i>minimum:</i> <code>1</code>)	<code>null</code>	The positional argument index, starting at 1. The index refers to position according to other positional argument. It does not define position in the argument list as a whole. When utilized with <code>multiple=true</code> , only the last positional argument may be defined as multiple (i.e. the one with the highest index).

BundleConfig

Configuration for tauri-bundler.

Type: `object`

Name	Type	Default	Description
<code>active</code>	<code>boolean</code>	<code>false</code>	Whether Tauri should bundle your application or just output the executable.
<code>targets</code>	<code>BundleTarget</code>	<code>view</code>	The bundle targets, currently supports ["deb", "appimage", "nsis", "msi", "app", "dmg", "updater"] or "all".
<code>identifier</code>	<code>string (required)</code>		The application identifier in reverse domain name notation (e.g. <code>com.tauri.example</code>). This string must be unique across applications since it is used in system configurations like the bundle ID and path to the webview data directory. This string must contain only

Name	Type	Default	Description
			alphanumeric characters (A-Z, a-z, and 0-9), hyphens (-), and periods (.)
publisher	string?	null	The application's publisher. Defaults to the second element in the identifier string. Currently maps to the Manufacturer property of the Windows Installer.
icon	string[]	[]	The app's icons
resources	BundleResources?	view	App resources to bundle. Each resource is a path to a file or directory. Glob patterns are supported.
copyright	string?	null	A copyright string associated with your application.
category	string?	null	<p>The application kind.</p> <p>Should be one of the following: Business, DeveloperTool, Education, Entertainment, Finance, Game, ActionGame, AdventureGame, ArcadeGame, BoardGame, CardGame, CasinoGame, DiceGame, EducationalGame, FamilyGame, KidsGame, MusicGame, PuzzleGame, RacingGame, RolePlayingGame, SimulationGame, SportsGame, StrategyGame, TriviaGame, WordGame, GraphicsAndDesign, HealthcareAndFitness, Lifestyle, Medical, Music, News, Photography, Productivity, Reference, SocialNetworking, Sports, Travel, Utility, Video, Weather.</p>
shortDescription	string?	null	A short description of your application.
longDescription	string?	null	A longer, multi-line description of the application.

Name	Type	Default	Description
appimage	AppImageConfig	view	Configuration for the AppImage bundle.
deb	DebConfig	view	Configuration for the Debian bundle.
macOS	MacConfig	view	Configuration for the macOS bundles.
externalBin	array?	null	<p>A list of—either absolute or relative—paths to binaries to embed with your application.</p> <p>Note that Tauri will look for system-specific binaries following the pattern "binary-name{-target-triple}{.system-extension}".</p> <p>E.g. for the external binary "my-binary", Tauri looks for:</p> <ul style="list-style-type: none"> - "my-binary-x86_64-pc-windows-msvc.exe" for Windows - "my-binary-x86_64-apple-darwin" for macOS - "my-binary-x86_64-unknown-linux-gnu" for Linux <p>so don't forget to provide binaries for all targeted platforms.</p>
windows	WindowsConfig	view	Configuration for the Windows bundle.

BundleTarget

Targets to bundle. Each value is case insensitive.

Can be any of the following types:

- "all": Bundle all targets.
- [BundleType](#): A list of bundle targets.
- [BundleType](#): A single bundle target.

BundleType

A bundle referenced by tauri-bundler.

Can be any **ONE** of the following types:

- "deb": The debian bundle (.deb).
- "appimage": The AppImage bundle (.appimage).
- "msi": The Microsoft Installer bundle (.msi).
- "nsis": The NSIS bundle (.exe).
- "app": The macOS application bundle (.app).
- "dmg": The Apple Disk Image bundle (.dmg).
- "updater": The Tauri updater bundle.

BundleResources

Definition for bundle resources. Can be either a list of paths to include or a map of source to target paths.

Can be any of the following types:

- `string[]`: A list of paths to include.
- `object`: A map of source to target paths.

AppImageConfig

Configuration for AppImage bundles.

Type: `object`

Name	Type	Default	Description
<code>bundleMediaFramework</code>	<code>boolean</code>	<code>false</code>	Include additional gstreamer dependencies needed for audio and video playback. This increases the bundle size by ~15-35MB depending on your build system.

DebConfig

Configuration for Debian (.deb) bundles.

Type: `object`

Name	Type	Default	Description
depends	array?	null	The list of deb dependencies your application relies on.
files	object	null	The files to include on the package.
desktopTemplate	string?	null	Path to a custom desktop file Handlebars template. Available variables: categories, comment (optional), exec, icon and name.

MacConfig

Configuration for the macOS bundles.

Type: object

Name	Type	Default	Description
frameworks	array?	null	A list of strings indicating any macOS X frameworks that need to be bundled with the application. If a name is used, ".framework" must be omitted and it will look for standard install locations. You may also use a path to a specific framework.
minimumSystemVersion	string?	null	A version string indicating the minimum macOS X version that the bundled application supports. Defaults to 10.13. Setting it to null completely removes the LSMinimumSystemVersion field on the bundle's Info.plist and the MACOSX_DEPLOYMENT_TARGET environment variable. An empty string is considered an invalid value so the default value is used.

Name	Type	Default	Description
exceptionDomain	string?	null	Allows your application to communicate with the outside world. It should be a lowercase, without port and protocol domain name.
license	string?	null	The path to the license file to add to the DMG bundle.
signingIdentity	string?	null	Identity to use for code signing.
providerShortName	string?	null	Provider short name for notarization.
entitlements	string?	null	Path to the entitlements file.

WindowsConfig

Windows bundler configuration.

Type: object

Name	Type	Default	Description
digestAlgorithm	string?	null	Specifies the file digest algorithm to use for creating file signatures. Required for code signing. SHA-256 is recommended.
certificateThumbprint	string?	null	Specifies the SHA1 hash of the signing certificate.
timestampUrl	string?	null	Server to use during timestamping.
tsp	boolean	false	Whether to use Time-Stamp Protocol (TSP, a.k.a. RFC 3161) for the timestamp server. Your code signing provider may use a TSP timestamp server, like e.g. SSL.com does. If so, enable TSP by setting to true.

Name	Type	Default	Description
webviewInstallMode	WebviewInstallMode	view	The installation mode for the Webview2 runtime.
webviewFixedRuntimePath	string?	null	<p>Path to the webview fixed runtime to use. Overwrites <code>webview_install_mode</code> if set.</p> <p>Will be removed in v2, prefer the <code>webview_install_mode</code> option.</p>
allowDowngrades	boolean	true	<p>Validates a second app installation, blocking the user from installing an older version if set to <code>false</code>.</p> <p>For instance, if <code>1.2.1</code> is installed, the user won't be able to install app version <code>1.2.0</code> or <code>1.1.5</code>.</p> <p>The default value of this flag is <code>true</code>.</p>
wix	WixConfig?	view	Configuration for the MSI generated with WiX.
nsis	NsisConfig?	view	Configuration for the installer generated with NSIS.

WebviewInstallMode

Install modes for the Webview2 runtime. Note that for the updater bundle `DownloadBootstrapper` is used.

For more information see <https://tauri.app/v1/guides/building/windows>.

Can be any **ONE** of the following types:

- { "type": "skip" }: Do not install the Webview2 as part of the Windows Installer.

Name	Type	Default	Description
type	"skip" (required)		undefined

- { "type": "downloadBootstrapper", "silent": boolean }: Download the bootstrapper and run it. Requires an internet connection. Results in a smaller installer size, but is not recommended on Windows 7.

Name	Type	Default	Description
type	"downloadBootstrapper" (required)		undefined
silent	boolean	true	Instructs the installer to run the bootstrapper in silent mode. Defaults to true.

- { "type": "embedBootstrapper", "silent": boolean }: Embed the bootstrapper and run it. Requires an internet connection. Increases the installer size by around 1.8MB, but offers better support on Windows 7.

Name	Type	Default	Description
type	"embedBootstrapper" (required)		undefined
silent	boolean	true	Instructs the installer to run the bootstrapper in silent mode. Defaults to true.

- { "type": "offlineInstaller", "silent": boolean }: Embed the offline installer and run it. Does not require an internet connection. Increases the installer size by around 127MB.

Name	Type	Default	Description
type	"offlineInstaller" (required)		undefined
silent	boolean	true	Instructs the installer to run the installer in silent mode. Defaults to true.

- { "type": "fixedRuntime", "path": string }: Embed a fixed webview2 version and use it at runtime. Increases the installer size by around 180MB.

Name	Type	Default	Description
type	"fixedRuntime" (required)		undefined
path	string (required)		The path to the fixed runtime to use. The fixed version can be downloaded on the official website . The .cab file must be extracted to a folder and this folder path must be defined on this field.

WixConfig

Configuration for the MSI bundle using WiX.

Type: object

Name	Type	Default	Description
language	WixLanguage	view	The installer languages to build. See https://docs.microsoft.com/en-us/windows/win32/msi/localizing-the-error-and-actiontext-tables .
template	string?	null	A custom .wxs template to use.
fragmentPaths	string[]	[]	A list of paths to .wxs files with WiX

Name	Type	Default	Description
			fragments to use.
componentGroupRefs	string[]	[]	The ComponentGroup element ids you want to reference from the fragments.
componentRefs	string[]	[]	The Component element ids you want to reference from the fragments.
featureGroupRefs	string[]	[]	The FeatureGroup element ids you want to reference from the fragments.
featureRefs	string[]	[]	The Feature element ids you want to reference from the fragments.
mergeRefs	string[]	[]	The Merge element ids you want to reference from the fragments.
skipWebviewInstall	boolean	false	<p>Disables the Webview2 runtime installation after app install.</p> <p>Will be removed in v2, prefer the <code>[WindowsConfig::webview_install_mode]</code> option.</p>
license	string?	null	<p>The path to the license file to render on the installer.</p> <p>Must be an RTF file, so if a different extension is provided, we convert it to the RTF format.</p>
enableElevatedUpdateTask	boolean	false	Create an elevated update task within Windows Task Scheduler.
bannerPath	string?	null	Path to a bitmap file to use as the installation user interface banner. This bitmap will appear at the top of all but the first page of the installer.

Name	Type	Default	Description
			The required dimensions are 493px × 58px.
dialogImagePath	string?	null	Path to a bitmap file to use on the installation user interface dialogs. It is used on the welcome and completion dialogs. The required dimensions are 493px × 312px.

WixLanguage

The languages to build using WiX.

Can be any of the following types:

- `string`: A single language to build, without configuration.
- `string[]`: A list of languages to build, without configuration.
- `WixLanguageConfig`: A map of languages and its configuration.

WixLanguageConfig

Configuration for a target language for the WiX build.

Type: `object`

Name	Type	Default	Description
localePath	string?	null	The path to a locale (.wxl) file. See https://wixtoolset.org/documentation/manual/v3/howtos/ui_and_localization/configuring_locales.html

NsisConfig

Configuration for the Installer bundle using NSIS.

Type: `object`

Name	Type	Default	
template	string?	null	A custom .nsi template to use.
license	string?	null	The path to the license file to render on the welcome screen.
headerImage	string?	null	The path to a bitmap file to display on the header bar. The recommended dimensions are 150px by 50px.
sidebarImage	string?	null	The path to a bitmap file for the Welcome screen sidebar. The recommended dimensions are 164px by 50px.
installerIcon	string?	null	The path to an icon file used as the installer icon.
installMode	NSISInstallerMode	view	Whether the installation will be for all users or just the current user.
languages	array?	null	A list of installer languages. By default the first language will be used. To allow the user to choose, see the NSIS documentation for the complete list of languages.
customLanguageFiles	object?	null	A key-value pair where the key is the language code and the value is the translated text for tauri's custom messages. See https://github.com/tauri-apps/tauri/blob/main/packages/tauri/src/i18n/languages/English.nsh for an example. Note: the key must be a valid NSIS language code.
displayLanguageSelector	boolean	false	Whether to display a language selector on the welcome screen. By default the OS language is selected, with the user able to change it.

Name	Type	Default	
compression	NsisCompression?	view	<p>Set the compression algorithm used to compress files.</p> <p>See https://nsis.sourceforge.io/Reference/SetCompressor</p>

NSISInstallerMode

Install Modes for the NSIS installer.

Can be any **ONE** of the following types:

- "currentUser": Default mode for the installer.

Install the app by default in a directory that doesn't require Administrator access.

Installer metadata will be saved under the `HKCU` registry path.

- "perMachine": Install the app by default in the `Program Files` folder directory requires Administrator access for the installation.

Installer metadata will be saved under the `HKLM` registry path.

- "both": Combines both modes and allows the user to choose at install time whether to install for the current user or per machine. Note that this mode will require Administrator access even if the user wants to install it for the current user only.

Installer metadata will be saved under the `HKLM` or `HKCU` registry path based on the user's choice.

NsisCompression

Compression algorithms used in the NSIS installer.

See <https://nsis.sourceforge.io/Reference/SetCompressor>

Can be any **ONE** of the following types:

- "zlib": ZLIB uses the deflate algorithm, it is a quick and simple method. With the default compression level it uses about 300 KB of memory.
- "bzip2": BZIP2 usually gives better compression ratios than ZLIB, but it is a bit slower and uses more memory. With the default compression level it uses about 4 MB of memory.

- "Izma": LZMA (default) is a new compression method that gives very good compression ratios. The decompression speed is high (10-20 MB/s on a 2 GHz CPU), the compression speed is lower. The memory size that will be used for decompression is the dictionary size plus a few KBs, the default is 8 MB.

AllowlistConfig

Allowlist configuration. The allowlist is a translation of the [Cargo allowlist features](#).

Notes

- Endpoints that don't have their own allowlist option are enabled by default.
- There is only "opt-in", no "opt-out". Setting an option to `false` has no effect.

Examples

- - `"app-all": true` will make the `hide` endpoint be available regardless of whether `hide` is set to `false` or `true` in the allowlist.

Type: `object`

Name	Type	Default	Description
<code>all</code>	<code>boolean</code>	<code>false</code>	Use this flag to enable all API features.
<code>fs</code>	<code>FsAllowlistConfig</code>	<code>view</code>	File system API allowlist.
<code>window</code>	<code>WindowAllowlistConfig</code>	<code>view</code>	Window API allowlist.
<code>shell</code>	<code>ShellAllowlistConfig</code>	<code>view</code>	Shell API allowlist.
<code>dialog</code>	<code>DialogAllowlistConfig</code>	<code>view</code>	Dialog API allowlist.
<code>http</code>	<code>HttpAllowlistConfig</code>	<code>view</code>	HTTP API allowlist.
<code>notification</code>	<code>NotificationAllowlistConfig</code>	<code>view</code>	Notification API allowlist.
<code>globalShortcut</code>	<code>GlobalShortcutAllowlistConfig</code>	<code>view</code>	Global shortcut API allowlist.

Name	Type	Default	Description
os	OsAllowlistConfig	view	OS allowlist.
path	PathAllowlistConfig	view	Path API allowlist.
protocol	ProtocolAllowlistConfig	view	Custom protocol allowlist.
process	ProcessAllowlistConfig	view	Process API allowlist.
clipboard	ClipboardAllowlistConfig	view	Clipboard APIs allowlist.
app	AppAllowlistConfig	view	App APIs allowlist.

FsAllowlistConfig

Allowlist for the file system APIs.

Type: object

Name	Type	Default	Description
scope	FsAllowlistScope	[]	The access scope for the filesystem APIs.
all	boolean	false	Use this flag to enable all file system API features.
readFile	boolean	false	Read file from local filesystem.
writeFile	boolean	false	Write file to local filesystem.
readDir	boolean	false	Read directory from local filesystem.
copyFile	boolean	false	Copy file from local filesystem.
createDir	boolean	false	Create directory from local filesystem.
removeDir	boolean	false	Remove directory from local filesystem.
removeFile	boolean	false	Remove file from local filesystem.

Name	Type	Default	Description
renameFile	boolean	false	Rename file from local filesystem.
exists	boolean	false	Check if path exists on the local filesystem.

FsAllowlistScope

Filesystem scope definition. It is a list of glob patterns that restrict the API access from the webview.

Each pattern can start with a variable that resolves to a system base directory. The variables are: \$AUDIO, \$CACHE, \$CONFIG, \$DATA, \$LOCALDATA, \$DESKTOP, \$DOCUMENT, \$DOWNLOAD, \$EXE, \$FONT, \$HOME, \$PICTURE, \$PUBLIC, \$RUNTIME, \$TEMPLATE, \$VIDEO, \$RESOURCE, \$APP, \$LOG, \$TEMP, \$APPCONFIG, \$APPDATA, \$APPOCALDATA, \$APPCACHE, \$APPLOG.

Can be any of the following types:

- string[]: A list of paths that are allowed by this scope.
- { "allow": string[], "deny": string[], "requireLiteralLeadingDot": boolean? }: A complete scope configuration.

Name	Type	Default	Description
allow	string[]	[]	A list of paths that are allowed by this scope.
deny	string[]	[]	A list of paths that are not allowed by this scope. This gets precedence over the Scope::allow list.
requireLiteralLeadingDot	boolean?	null	Whether or not paths that contain components that start with a . will require that . appears literally in the pattern; *, ?, **, or [...] will not match. This is useful because such files are conventionally considered hidden on Unix systems and it might be desirable to skip them when listing files.

Name	Type	Default	Description
			Defaults to <code>true</code> on Unix systems and <code>false</code> on Windows

WindowAllowlistConfig

Allowlist for the window APIs.

Type: object

Name	Type	Default	Description
<code>all</code>	boolean	<code>false</code>	Use this flag to enable all window API features.
<code>create</code>	boolean	<code>false</code>	Allows dynamic window creation.
<code>center</code>	boolean	<code>false</code>	Allows centering the window.
<code>requestUserAttention</code>	boolean	<code>false</code>	Allows requesting user attention on the window.
<code>setResizable</code>	boolean	<code>false</code>	Allows setting the resizable flag of the window.
<code>setMaximizable</code>	boolean	<code>false</code>	Allows setting whether the window's native maximize button is enabled or not.
<code>setMinimizable</code>	boolean	<code>false</code>	Allows setting whether the window's native minimize button is enabled or not.
<code>setClosable</code>	boolean	<code>false</code>	Allows setting whether the window's native close button is enabled or not.
<code>setTitle</code>	boolean	<code>false</code>	Allows changing the window title.
<code>maximize</code>	boolean	<code>false</code>	Allows maximizing the window.
<code>unmaximize</code>	boolean	<code>false</code>	Allows unmaximizing the window.
<code>minimize</code>	boolean	<code>false</code>	Allows minimizing the window.

Name	Type	Default	Description
unminimize	boolean	false	Allows unminimizing the window.
show	boolean	false	Allows showing the window.
hide	boolean	false	Allows hiding the window.
close	boolean	false	Allows closing the window.
setDecorations	boolean	false	Allows setting the decorations flag of the window.
setAlwaysOnTop	boolean	false	Allows setting the always_on_top flag of the window.
setContentProtected	boolean	false	Allows preventing the window contents from being captured by other apps.
setSize	boolean	false	Allows setting the window size.
setMinSize	boolean	false	Allows setting the window minimum size.
setMaxSize	boolean	false	Allows setting the window maximum size.
setPosition	boolean	false	Allows changing the position of the window.
setFullscreen	boolean	false	Allows setting the fullscreen flag of the window.
setFocus	boolean	false	Allows focusing the window.
setIcon	boolean	false	Allows changing the window icon.
setSkipTaskbar	boolean	false	Allows setting the skip_taskbar flag of the window.
setCursorGrab	boolean	false	Allows grabbing the cursor.
setCursorVisible	boolean	false	Allows setting the cursor visibility.
setCursorIcon	boolean	false	Allows changing the cursor icon.

Name	Type	Default	Description
setCursorPosition	boolean	false	Allows setting the cursor position.
setIgnoreCursorEvents	boolean	false	Allows ignoring cursor events.
startDragging	boolean	false	Allows start dragging on the window.
print	boolean	false	Allows opening the system dialog to print the window content.

ShellAllowlistConfig

Allowlist for the shell APIs.

Type: object

Name	Type	Default	Description
scope	ShellAllowlistScope	[]	Access scope for the binary execution APIs. Sidecars are automatically enabled.
all	boolean	false	Use this flag to enable all shell API features.
execute	boolean	false	Enable binary execution.
sidecar	boolean	false	Enable sidecar execution, allowing the JavaScript layer to spawn a sidecar command, an executable that is shipped with the application. For more information see https://tauri.app/v1/guides/building/sidecar .
open	ShellAllowlistOpen	false	Open URL with the user's default application.

ShellAllowlistScope

Shell scope definition. It is a list of command names and associated CLI arguments that restrict the API access from the webview.

Type: ShellAllowedCommand

ShellAllowedCommand

A command allowed to be executed by the webview API.

Type: object

Name	Type	Default	Description
name	string (required)		<p>The name for this allowed shell command configuration.</p> <p>This name will be used inside of the webview API to call this command along with any specified arguments.</p>
cmd	string	null	<p>The command name. It can start with a variable that resolves to a system base directory. The variables are:</p> <p>\$AUDIO, \$CACHE, \$CONFIG, \$DATA, \$LOCALDATA, \$DESKTOP, \$DOCUMENT, \$DOWNLOAD, \$EXE, \$FONT, \$HOME, \$PICTURE, \$PUBLIC, \$RUNTIME, \$TEMPLATE, \$VIDEO, \$RESOURCE, \$APP, \$LOG, \$TEMP, \$APPCONFIG, \$APPDATA, \$APPLOCALDATA, \$APPCACHE, \$APPLOG.</p>
args	ShellAllowedArgs	false	The allowed arguments for the command execution.
sidecar	boolean	false	If this command is a sidecar command.

ShellAllowedArgs

A set of command arguments allowed to be executed by the webview API.

A value of true will allow any arguments to be passed to the command. false will disable all arguments. A list of [ShellAllowedArg] will set those arguments as the only valid arguments to be passed to the attached command configuration.

Can be any of the following types:

- boolean: Use a simple boolean to allow all or disable all arguments to this command configuration.
- ShellAllowedArg: A specific set of [ShellAllowedArg] that are valid to call for the command configuration.

ShellAllowedArg

A command argument allowed to be executed by the webview API.

Can be any of the following types:

- `string`: A non-configurable argument that is passed to the command in the order it was specified.
- `{ "validator": string }`: A variable that is set while calling the command from the webview API.

Name	Type	Default	Description
<code>validator</code>	<code>string</code> (required)		<p><code>regex</code> validator to require passed values to conform to an expected input.</p> <p>This will require the argument value passed to this variable to match the <code>validator</code> <code>regex</code> before it will be executed.</p>

ShellAllowlistOpen

Defines the `shell > open` api scope.

Can be any of the following types:

- `boolean`: If the shell open API should be enabled.

If enabled, the default validation regex (`^((mailto:\w+)|(tel:\w+)|(https?://\w+)).+`) is used.

- `string`: Enable the shell open API, with a custom regex that the opened path must match against.

If using a custom regex to support a non-`http(s)` schema, care should be used to prevent values that allow flag-like strings to pass validation. e.g. `--enable-debugging`, `-i`, `/R`.

DialogAllowlistConfig

Allowlist for the dialog APIs.

Type: `object`

Name	Type	Default	Description
<code>all</code>	<code>boolean</code>	<code>false</code>	Use this flag to enable all dialog API features.

Name	Type	Default	Description
open	boolean	false	Allows the API to open a dialog window to pick files.
save	boolean	false	Allows the API to open a dialog window to pick where to save files.
message	boolean	false	Allows the API to show a message dialog window.
ask	boolean	false	Allows the API to show a dialog window with Yes/No buttons.
confirm	boolean	false	Allows the API to show a dialog window with Ok/Cancel buttons.

HttpAllowlistConfig

Allowlist for the HTTP APIs.

Type: object

Name	Type	Default	Description
scope	HttpAllowlistScope	[]	The access scope for the HTTP APIs.
all	boolean	false	Use this flag to enable all HTTP API features.
request	boolean	false	Allows making HTTP requests.

HttpAllowlistScope

HTTP API scope definition. It is a list of URLs that can be accessed by the webview when using the HTTP APIs. The scoped URL is matched against the request URL using a glob pattern.

Examples:

- "https://*": allows all HTTPS urls
- "https://*.github.com/tauri-apps/tauri": allows any subdomain of "github.com" with the "tauri-apps/api" path
- "https://myapi.service.com/users/*": allows access to any URLs that begins with "https://myapi.service.com/users/"

Type: string _(format: uri)_[]

NotificationAllowlistConfig

Allowlist for the notification APIs.

Type: object

Name	Type	Default	Description
all	boolean	false	Use this flag to enable all notification API features.

GlobalShortcutAllowlistConfig

Allowlist for the global shortcut APIs.

Type: object

Name	Type	Default	Description
all	boolean	false	Use this flag to enable all global shortcut API features.

OsAllowlistConfig

Allowlist for the OS APIs.

Type: object

Name	Type	Default	Description
all	boolean	false	Use this flag to enable all OS API features.

PathAllowlistConfig

Allowlist for the path APIs.

Type: object

Name	Type	Default	Description
all	boolean	false	Use this flag to enable all path API features.

ProtocolAllowlistConfig

Allowlist for the custom protocols.

Type: object

Name	Type	Default	Description
assetScope	FsAllowlistScope	[]	The access scope for the asset protocol.
all	boolean	false	Use this flag to enable all custom protocols.
asset	boolean	false	Enables the asset protocol.

ProcessAllowlistConfig

Allowlist for the process APIs.

Type: object

Name	Type	Default	Description
all	boolean	false	Use this flag to enable all process APIs.
relaunch	boolean	false	Enables the relaunch API.
relaunchDangerousAllowSymlinkMacos	boolean	false	Dangerous option that allows macOS to relaunch even if the binary contains a symlink. This is due to macOS having less symlink protection. Highly recommended to not set this flag unless you have a very specific reason too, and understand the implications of it.
exit	boolean	false	Enables the exit API.

ClipboardAllowlistConfig

Allowlist for the clipboard APIs.

Type: object

Name	Type	Default	Description
all	boolean	false	Use this flag to enable all clipboard APIs.
writeText	boolean	false	Enables the clipboard's <code>writeText</code> API.
readText	boolean	false	Enables the clipboard's <code>readText</code> API.

AppAllowlistConfig

Allowlist for the app APIs.

Type: object

Name	Type	Default	Description
all	boolean	false	Use this flag to enable all app APIs.
show	boolean	false	Enables the app's <code>show</code> API.
hide	boolean	false	Enables the app's <code>hide</code> API.

SecurityConfig

Security configuration.

Type: object

Name	Type	Default	Description
csp	Csp?	view	The Content Security Policy will be injected on the built app if <code>dev_csp</code> is not set.

Name	Type	Default	Desc
			value is also injected into the configuration. This is a really important configuration because you ensure your assets are secured. See https://develop-us/docs/Web/Security/CSP
devCsp	Csp?	view	The Content Security Policy will be injected into the configuration when developing. This is a really important configuration because you ensure your assets are secured. See https://develop-us/docs/Web/Security/CSP
freezePrototype	boolean	false	Freezes the Object prototype when using the <code>Object.create</code> function.
dangerousDisableAssetCspModification	DisabledCspModificationKind	false	Disables the Tampermonkey CSP modification sources. At compile time, the frontend asset loader changes the Content Security Policy to only allow your own script sources. This stops sources which may introduce XSS when using alone or flexing sources.
			This configuration is used to disable the CSP modification for assets.

Name	Type	Default	Desc
			<p>both a boolean strings as value instructs Tauri to inject for all and a list of string CSP directives to inject.</p> <p>WARNING: Only know what you have properly configured CSP. Your application is vulnerable to XSS without this Tauri configuration.</p>
dangerousRemoteDomainIpcAccess	RemoteDomainAccessScope	[]	<p>Allow external commands to Tauri to access the ipc access of a remote domain.</p> <p>By default, external domains do not have access to the ipc access of a remote domain. They cannot control the commands sent to the ipc access. This prevents a malicious externally loaded script from compromised scripts from executing commands on the user's device.</p> <p>This configuration allows external domains to access the ipc access of a remote domain. You can configure a list of allowed to access domains. Subpaths are also allowed. Subdomains are not allowed.</p> <p>WARNING: Only know what you have properly configured CSP. Your application is vulnerable to XSS without this Tauri configuration.</p>

Name	Type	Default	Desc
			you either have against malicious or you can trust external sites. You might be vulnerable dangerous Taur related attacks
dangerousUseHttpScheme	boolean	false	<p>Sets whether the browser protocols should use <code>http://<scheme></code> instead of the default <code>https://<scheme></code>. Windows.</p> <p>WARNING: Using this setting will allow mixed content trying to fetch resources and is therefore not safe. It will match the <code><scheme>://loc</code> used on macOS.</p>

Csp

A Content-Security-Policy definition. See <https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>.

Can be any of the following types:

- `string`: The entire CSP policy in a single text string.
- `CspDirectiveSources`: An object mapping a directive with its sources values as a list of strings.

CspDirectiveSources

A Content-Security-Policy directive source list. See <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/Sources#sources>.

Can be any of the following types:

- `string`: An inline list of CSP sources. Same as `List`, but concatenated with a space separator.
- `string[]`: A list of CSP sources. The collection will be concatenated with a space separator for the CSP string.

DisabledCspModificationKind

The possible values for the `dangerous_disable_asset_csp_modification` config option.

Can be any of the following types:

- `boolean`: If `true`, disables all CSP modification. `false` is the default value and it configures Tauri to control the CSP.
- `string[]`: Disables the given list of CSP directives modifications.

RemoteDomainAccessScope

External command access definition.

Type: `object`

Name	Type	Default	Description
<code>scheme</code>	<code>string?</code>	<code>null</code>	The URL scheme to allow. By default, all schemas are allowed.
<code>domain</code>	<code>string (required)</code>		The domain to allow.
<code>windows</code>	<code>string[] (required)</code>		The list of window labels this scope applies to.
<code>plugins</code>	<code>string[]</code>	<code>[]</code>	The list of plugins that are allowed in this scope. The names should be without the <code>tauri-plugin-</code> prefix, for example "store" for <code>tauri-plugin-store</code> .
<code>enableTauriAPI</code>	<code>boolean</code>	<code>false</code>	Enables access to the Tauri API.

UpdaterConfig

The Updater configuration object.

Type: object

Name	Type	Default	Description
active	boolean	false	Whether the updater is active or not.
dialog	boolean	true	Display built-in dialog or use event system if disabled.
endpoints	UpdaterEndpoint ?	null	<p>The updater endpoints. TLS is enforced on production.</p> <p>The updater URL can contain the following variables:</p> <ul style="list-style-type: none">- {{current_version}}: The version of the app that is requesting the update- {{target}}: The operating system name (one of linux, windows or darwin).- {{arch}}: The architecture of the machine (one of x86_64, i686, aarch64 or armv7). <p># Examples</p> <ul style="list-style-type: none">- "https://my.cdn.com/latest.json": a raw JSON endpoint that returns the latest version and download links for each platform.- "https://updates.app.dev/{{target}}?version={{current_version}}&arch={{arch}}": a dedicated API with positional and query string arguments.
pubkey	string	null	Signature public key.
windows	UpdaterWindowsConfig	view	The Windows configuration for the updater.

UpdaterEndpoint

A URL to an updater server.

The URL must use the https scheme on production.

Type: string (format: uri)

UpdaterWindowsConfig

The updater configuration for Windows.

Type: object

Name	Type	Default	Description
installerArgs	string[]	[]	Additional arguments given to the NSIS or WiX installer.
installMode	WindowsUpdateInstallMode	view	The installation mode for the update on Windows. Defaults to passive.

WindowsUpdateInstallMode

Install modes for the Windows update.

Can be any **ONE** of the following types:

- "basicUI": Specifies there's a basic UI during the installation process, including a final dialog box at the end.
- "quiet": The quiet mode means there's no user interaction required. Requires admin privileges if the installer does.
- "passive": Specifies unattended mode, which means the installation only shows a progress bar.

SystemTrayConfig

Configuration for application system tray icon.

Type: object

Name	Type	Default	Description
iconPath	string (required)		Path to the default icon to use on the system tray.
iconAsTemplate	boolean	false	A Boolean value that determines whether the image represents a template image on macOS.

Name	Type	Default	Description
menuOnLeftClick	boolean	true	A Boolean value that determines whether the menu should appear when the tray icon receives a left click on macOS.
title	string?	null	Title for MacOS tray

BuildConfig

The Build configuration object.

Type: object

Name	Type	Default	Description
runner	string?	null	The binary used to build and run the application.
devPath	AppUrl	view	<p>The path to the application assets or URL to load in development.</p> <p>This is usually an URL to a dev server, which serves your application assets with live reloading. Most modern JavaScript bundlers provides a way to start a dev server by default.</p> <p>See vite, Webpack DevServer and sirv for examples on how to set up a dev server.</p>
distDir	AppUrl	view	<p>The path to the application assets or URL to load in production.</p> <p>When a path relative to the configuration file is provided, it is read recursively and all files are embedded in the application binary. Tauri then looks for an <code>index.html</code> file unless you provide a custom window</p>

Name	Type	Default	Description
			<p>URL.</p> <p>You can also provide a list of paths to be embedded, which allows granular control over what files are added to the binary. In this case, all files are added to the root and you must reference it that way in your HTML files.</p> <p>When an URL is provided, the application won't have bundled assets and the application will load that URL by default.</p>
beforeDevCommand	BeforeDevCommand?	view	<p>A shell command to run before <code>tauri dev</code> kicks in.</p> <p>The TAURI_PLATFORM, TAURI_ARCH, TAURI_FAMILY, TAURI_PLATFORM_VERSION, TAURI_PLATFORM_TYPE and TAURI_DEBUG environment variables are set if you perform conditional compilation.</p>
beforeBuildCommand	HookCommand?	view	<p>A shell command to run before <code>tauri build</code> kicks in.</p> <p>The TAURI_PLATFORM, TAURI_ARCH, TAURI_FAMILY, TAURI_PLATFORM_VERSION, TAURI_PLATFORM_TYPE and TAURI_DEBUG environment variables are set if you perform conditional compilation.</p>
beforeBundleCommand	HookCommand?	view	<p>A shell command to run before the bundling phase in <code>tauri build</code> kicks in.</p> <p>The TAURI_PLATFORM, TAURI_ARCH, TAURI_FAMILY,</p>

Name	Type	Default	Description
			TAURI_PLATFORM_VERSION, TAURI_PLATFORM_TYPE and TAURI_DEBUG environment variables are set if you perform conditional compilation.
features	array?	null	Features passed to cargo commands.
withGlobalTauri	boolean	false	Whether we should inject the Tauri API on window.__TAURI__ or not.

AppUrl

Defines the URL or assets to embed in the application.

Can be any of the following types:

- WindowUrl : The app's external URL, or the path to the directory containing the app assets.
- string[] : An array of files to embed on the app.

BeforeDevCommand

Describes the shell command to run before tauri dev.

Can be any of the following types:

- string : Run the given script with the default options.
- { "script": string, "cwd": string?, "wait": boolean } : Run the given script with custom options.

Name	Type	Default	Description
script	string (required)		The script to execute.
cwd	string?	null	The current working directory.
wait	boolean	false	Whether tauri dev should wait for the command to finish or not. Defaults to false.

HookCommand

Describes a shell command to be executed when a CLI hook is triggered.

Can be any of the following types:

- `string`: Run the given script with the default options.
- `{ "script": string, "cwd": string? }`: Run the given script with custom options.

Name	Type	Default	Description
<code>script</code>	string (required)		The script to execute.
<code>cwd</code>	string?	<code>null</code>	The current working directory.

PluginConfig

The plugin configs holds a HashMap mapping a plugin name to its configuration object.

Type: `object`

 [Edit this page](#)

Last updated on Sep 30, 2023