Frequently Asked Questions

How can I use unpublished Tauri changes?

To use Tauri from GitHub (bleeding edge version) you need to change your `Cargo.toml` file and update your CLI and API.

> ▶ Pulling the Rust crate from source

> ▶ Using the Tauri CLI from source

> ▶ Using the Tauri API from source

Should I use Node or Cargo?

Even though installing the CLI through Cargo is the preferred option, it has to compile the whole binary from scratch when you install it. If you're in a CI environment or on a very slow machine you're better off choosing another installation method.

As the CLI is written in Rust, it is naturally available through crates.io and installable with Cargo.

We also compile the CLI as a native Node.js addon and distribute it via npm. This has several advantages compared to the Cargo installation method:

1. The CLI is pre-compiled, leading to much faster install times
2. You can pin a specific version in your package.json file
3. If you develop custom tooling around Tauri, you can import the CLI as a regular JavaScript module
4. You can install the CLI using a JavaScript manager

Recommended Browserlist

We recommend using `es2021`, `last 3 Chrome versions`, and `safari 13` for your browserlist and build targets. Tauri leverages the OS's native rendering engine (WebKit on macOS, WebView2 on Windows and WebKitGTK on Linux).

Build Conflict with Homebrew on Linux

Homebrew on Linux includes its own `pkg-config` (a utility to find libraries on the system). This can cause conflicts when installing the same `pkg-config` package for Tauri (usually installed through the package manager like `apt`). When you try to build a Tauri app it will try to invoke `pkg-config` and will end up invoking the one from Homebrew. If Homebrew wasn't used to install Tauri's dependencies, this can cause errors.

Errors will *usually* contain messages along the lines of `error: failed to run custom build command for X` - `Package Y was not found in the pkg-config search path.`. Note that you may see similar errors if the required dependencies are not installed at all.

There are two solutions to this issue:

1. Uninstall Homebrew
2. Set the `PKG_CONFIG_PATH` environment variable to point to the correct `pkg-config` before building a Tauri app

✏ Edit this page

Last updated on Aug 18, 2023