Features

Making Your Own CLI

Tauri enables your app to have a CLI through clap, a robust command line argument parser. With a simple CLI definition in your tauri.conf.json file, you can define your interface and read its argument matches map on JavaScript and/or Rust.

Calling Rust from the frontend

Tauri provides a simple yet powerful command system for calling Rust functions from your web app.

Events

The Tauri event system is a multi-producer multi-consumer communication primitive that allows message passing between the frontend and the backend.

Icons

Tauri ships with a default iconset based on its logo. This is NOT what you want when you ship your application. To remedy this common situation, Tauri provides the icon command that will take an input file ("./app-icon.png" by default) and create all the icons needed for the various platforms.

Window Menu

Native application menus can be attached to a window.

📄 Multiwindow

Manage multiple windows on a single application.

📄 Tauri Plugins

Plugins allow you to hook into the Tauri application lifecycle and introduce new commands.

📄 Splashscreen

If your webpage could take some time to load, or if you need to run an initialization procedure in Rust before displaying your main window, a splashscreen could improve the loading experience for the user.

📄 System Tray

Native application system tray.

📄 Window Customization

Tauri provides lots of options for customizing the look and feel of your app's window. You can create custom titlebars, have transparent windows, enforce size constraints, and more.

Edit this page

*Last updated on **Jun 15, 2022***