

# TauriでカスタマイズしたTODOアプリを作ってみる



Press Space for next page →





# 自己紹介

- 📝 飯野陽平 ([wheatandcat](https://www.wheatandcat.me/))
- 🏢 フリーランスエンジニア (シェアフル株式会社CTO)
- 💻 Blog: <https://www.wheatandcat.me/>
- 🛠️ 今までに作ったもの
  - [memoir](#)
  - [ペペロミア](#)
  - [Atomic Design Check List](#)

# Tauriとは

- TauriはRustで作成されたクロスプラットフォームでデスクトップアプリが作成できるフレームワーク
- デスクトップアプリのネイティブ部分のコードはRust、フロントエンドはWebの技術を使用できる
  - なのでフロントエンドに関しては以下を使用できる
    - HTML
    - React
    - Next.js
    - Svelte
- 開発の領域的には、Electronと同じ
- [Tauri](#)
- [Rust](#)

# Electronとの比較

- 違いについては以下を参照
  - [Tauri 1.0が正式リリース！概要や特徴、Electronとの違いを解説 | アンドエンジニア](#)
- 主な違い以下の部分
  - Electronは、ChromiumをそのままバンドルしているがWebViewエンジンの[wry](#)を使用している
  - Rustで実装されており、バンドルサイズ、消費CPU、セキュリティ共に改善されている
- [Electron](#)

# TODOアプリを作ってみた

- Tauriを使用してTODOアプリを作ってみた。
- GitHub
  - <https://github.com/wheatandcat/todo>
- 使用技術
  - [Tauri](#)
  - [React](#)
  - [markdown-to-jsx](#)
  - [Tailwind CSS](#)



# モチベーション

- 仕事のTODOリストは今までslackのDMに書いて運用していた
  - 雑な運用でリストで書き出して、タスクが終わったら「✓」をつけるのみで運用
  - 他のTODOアプリも試したが無駄にリッチなアプリが多くて続かなかった
  - ブラウザだと、他の作業で間違って消したりするのでデスクトップアプリで作りたかった
  - 仕事のTODOだとセンシティブな内容が多かったので、サーバー通信は避けたい
- なので、自分用にカスタマイズしたTODOアプリを作ってみた

# Tauriでの開発①

以下でブラウザでローカル環境を起動

```
$ yarn dev
```

これだとブラウザで起動するので、ブラウザのDevToolも活用できる。

# Tauriでの開発②

以下でデスクトップアプリでローカル環境を起動

```
$ yarn tauri dev
```

これだとデスクトップアプリの状態で起動するので、ネイティブの機能をデバッグする場合は、こちらで起動。

ホットリロードも有効なので、快適に開発できる。



# Tauriでの開発③

以下でデスクトップアプリをビルド

```
$ yarn tauri build
```

デフォルトだとhostのデバイの環境で起動できるアプリにビルド。

以下のオプションで各バイナリに変換可能。

- [binary options](#)

# frontendの開発

- frontendはReactで作成
  - [コード](#)
- Markdownの表示は[markdown-to-jsx](#)を使用
  - [コード](#)
- Markdownのparseは[remark-parse](#)を使用
  - checkbox部分のパースする自前で作成
  - [コード](#)

# ネイティブ機能の開発

- アプリのmenuは以下で作成
  - [コード](#)
  - 詳細は、[こちら](#)
- menuからfrontend側の通信は、以下のように作成
  - ネイティブから送信
    - [コード](#)
  - frontendで受け取り
    - [コード](#)
    - frontend側から、Tauri APIを使用してネイティブの機能も活用可能
      - [@tauri-apps/api | Tauri Apps](#)

# まとめ

- [Tauri](#)での開発は快適
- React経験者なら、**ほぼ学習コスト無しで作れる**
- ネイティブ機能の部分が[Rust](#)なので書きやすい
  - ビルドが通れば、ほぼOKな感じの安心感がある
- ちゃんと開発できたらApple/Windowsストアで公開する予定

**ご清聴ありがとうございました**