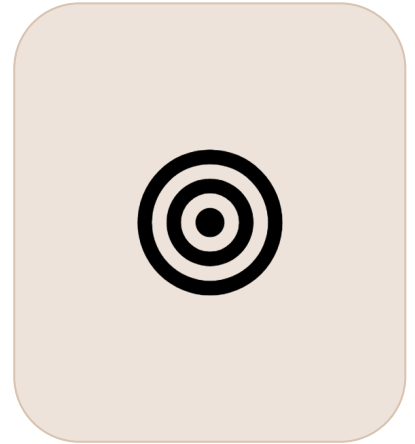
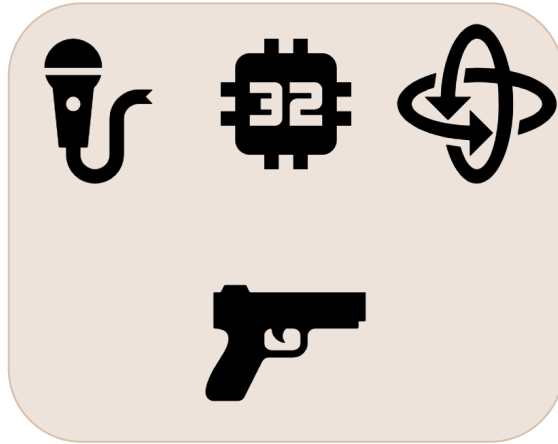
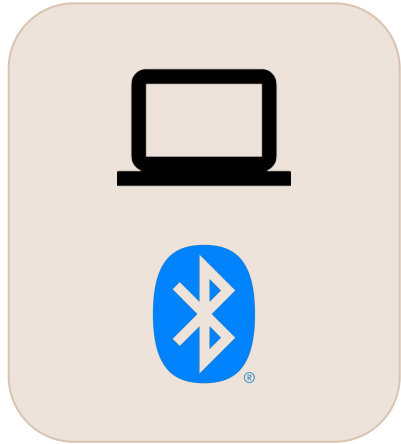


Création d'applications cross-plateforme avec Tauri

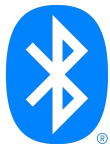
Clément FORGEARD
RennesJS - 21 mars 2024



Le projet



Début du développement



WebBluetooth

	Chrome	Edge	Firefox	Opera	Safari
					
Bluetooth 	 56	 79	 No	 43	 No

Le support Linux n'est pas activé par défaut

Migration vers Electron

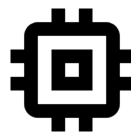


Application prête ! mais...



~ 270 MB

Taille code : 1 MB



~ 400 MB



Faibles de
sécurité

```
<script src="https://example.com/analytics.js"></script>
```



La solution : Tauri



1.0 sorti en juin 2022
76k stars
350 sponsors (60k\$)

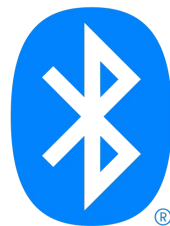


+



Cargo

+



btleplug

Compatible avec n'importe quel framework/sans framework

Migration de l'application Angular vers Tauri

```
$ npx tauri init
```

- ✓ What is your app name? · airsoft-training
- ✓ What should the window title be? · Airsoft Training
- ✓ Where are your web assets (HTML/CSS/JS) located ? · [...]
- ✓ What is the url of your dev server? · http://localhost:4200
- ✓ What is your frontend dev command? · npm run start
- ✓ What is your frontend build command? · npm run build

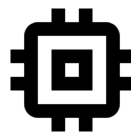
+ migration du code NodeJS en Rust 🙄

Avec <https://doc.rust-lang.org/rust-by-example/> + ChatGPT + Copilot + docs

Une amélioration très visible



~ 12 MB
(-180%)



~ 130 MB
(-100%)



Pas de failles
de sécurité

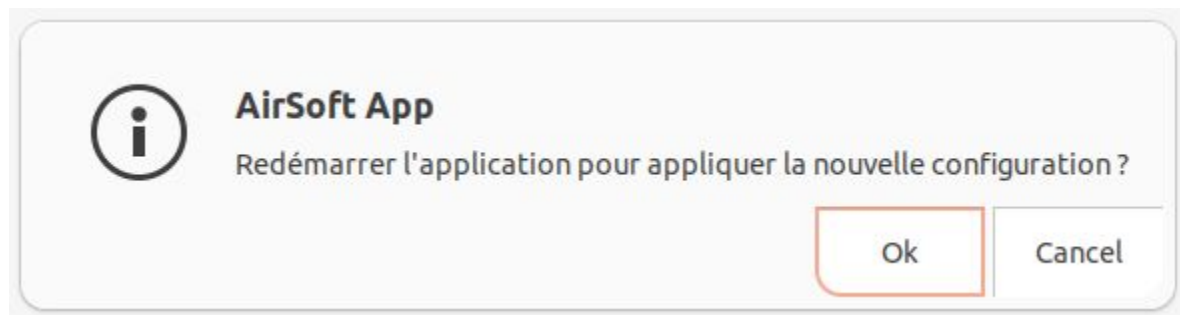
Deux applications en une : JS & Rust

- ▼ src
 - ▼ app
 - # app.component.css
 - <> app.component.html
 - TS app.component.ts
 - TS app.config.ts
 - TS app.routes.ts
 - > assets
 - <> index.html
 - TS main.ts
 - # styles.css

- ▼ src-tauri
 - > icons
 - ▼ src
 - Ⓡ main.rs
 - > target
 - ◆ .gitignore
 - Ⓡ build.rs
 - ≡ Cargo.lock
 - ⚙ Cargo.toml
 - { } tauri.conf.json

Gestion de la configuration

Redémarrage lors du changement de la configuration



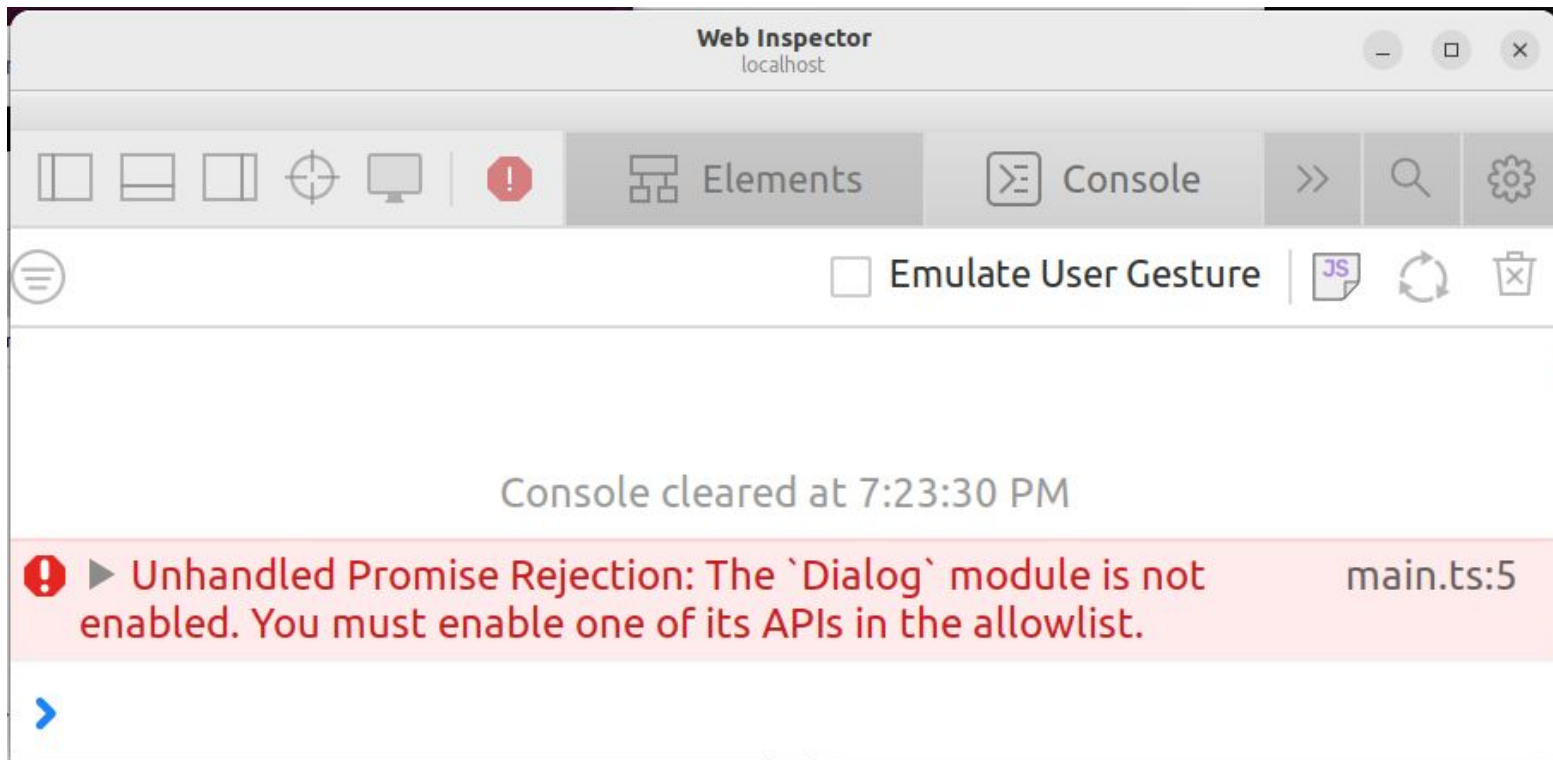
Redémarrage lors du changement de la configuration

```
import { confirm } from '@tauri-apps/api/dialog';
import { relaunch } from '@tauri-apps/api/process';

async function onConfigChanged() {
  const relaunchNow = await confirm(
    "Redémarrer l'application pour appliquer la nouvelle configuration ?",
    "AirSoft App"
  )
  if (relaunchNow) {
    await relaunch();
  }
}
```



Rien ne se passe...



Accès aux APIs optin, toutes refusés par défaut

```
{  
  "tauri": {  
    "allowlist": {  
      "dialog": {  
        "confirm": true  
      }  
    }  
  }  
}
```



Stockage de la configuration

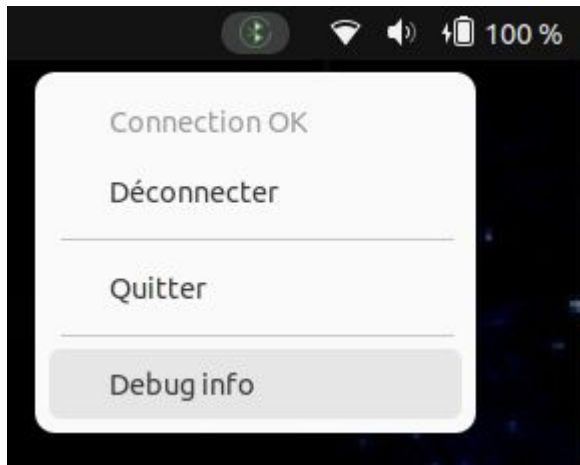
Error: path not allowed on the configured scope:
/home/cforgeard/.config/airsoft.training/config.json

```
{ "allowlist": {  
  // $CONFIG -> dossier de configuration (dépend de l'OS)  
  "fs": {  
    "scope": [ "$CONFIG/*" ],  
    "writeFile": true  
  }  
}}
```



Gestion de l'état de la connection Bluetooth

Etat & contrôle de l'application



```
let quit = CustomMenuItem::new("quit".to_string(), "Quitter");
```



Affichage de l'état de la connexion

```
app.emit_all("connection-state-change",  
  Payload { message: "connected".into() }).unwrap();  
//le payload doit pouvoir être sérialisé en JSON
```



JSON ↓ {...}

```
import { listen } from '@tauri-apps/api/event'  
listen('connection-state-change', (evt) => {  
  const isConnected = evt.payload === 'connected'  
}))
```



Bouton 'déconnecter'

```
#[tauri::command]
fn disconnect(reason: String) -> String {
    println!("Disconnect, reason: {}", reason);
}
```



```
import { invoke } from '@tauri-apps/api/tauri'
invoke('disconnect', { reason: 'user-request' })
    .then(() => console.log('disconnect OK'))
```



Activation/désactivation bluetooth sous Linux

```
import { Command } from '@tauri-apps/api/shell'  
new Command('systemctl', ['start', 'bluetooth'])
```



```
{  
  "scope": [{  
    "name": "switch-on-off-bluetooth",  
    "cmd": "systemctl",  
    "args": [{ "validator": "^start|stop$" }, "bluetooth"]  
  ]  
}
```



Export de statistiques vers un serveur distant

```
import { getClient } from '@tauri-apps/api/http';  
// via JS (limité par CORS + CSP)  
// exemple CSP: " default-src 'self'; img-src https://* "  
fetch('[...]');  
  
// via Rust (besoin de spécifier les URLs autorisés dans tauri.conf)  
const client = await getClient();  
client.request({method: 'POST', url: '[...]'});
```



Tests Rust/JS



```
import { mockIPC } from "@tauri-apps/api/mocks";  
test("disconnect button", async () => {  
  mockIPC((cmd, args) => { // permet de mocker invoke('disconnect')  
    if(cmd === "disconnect") {return undefined;}  
  });  
});
```



Publication

Installateur

Installateur
.exe/.msi

Installateur
.dmg

Paquet logiciel
.deb / .rpm

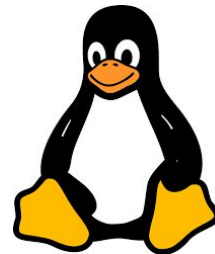
Application
standalone

Application .exe

macOS
Application
Bundle .app

AppImage

PAS de
cross-compilation
(besoin CI)



Système de mise à jour

```
{ "version": "v1.0.0",  
  "notes": "Airsoft Training",  
  "pub_date":  
    "2020-06-22T19:25:57Z",  
  "platforms": {  
    "darwin-x86_64": {  
      "signature": "...",  
      "url": "..."  
    }, [...]  
  } }
```

Peut être
hébergé sur
GitHub

Débogage – Rust

Debugger



CodeLLDB

152ms

A native debugger powered by ...

Vadim Chugunov



Autocomplétion / refactoring



rust-analyzer

41ms

Rust language support for Visu...

The Rust Programming La...

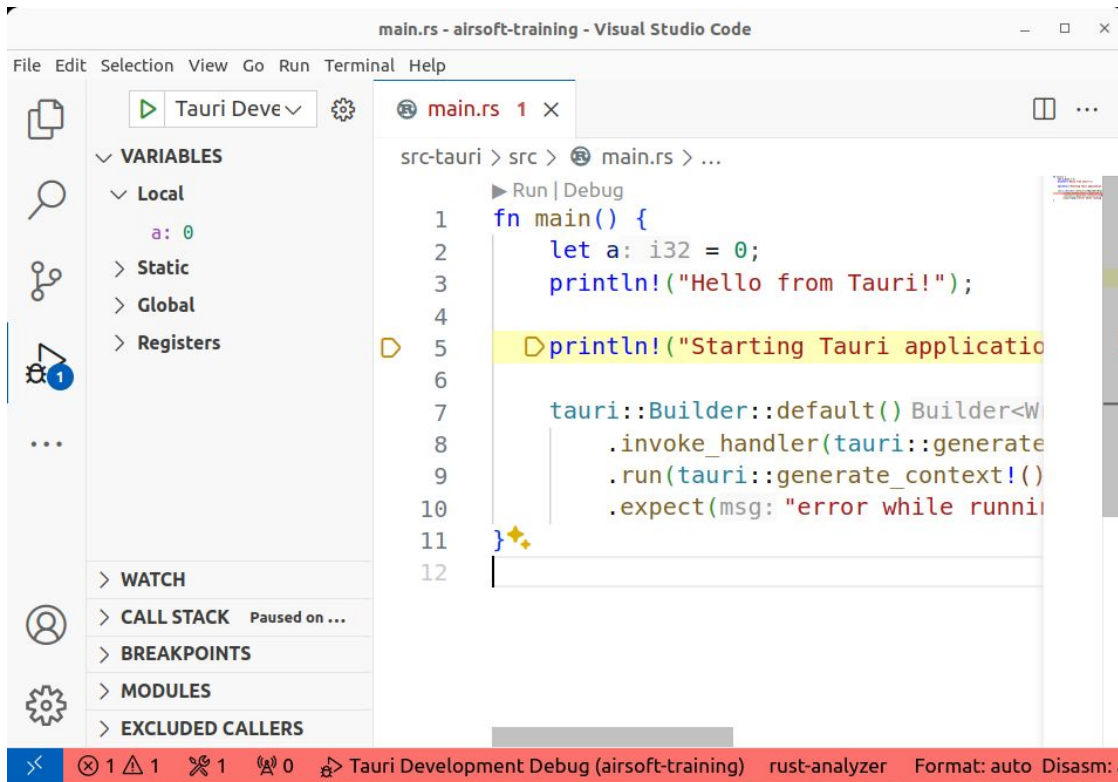


Tauri

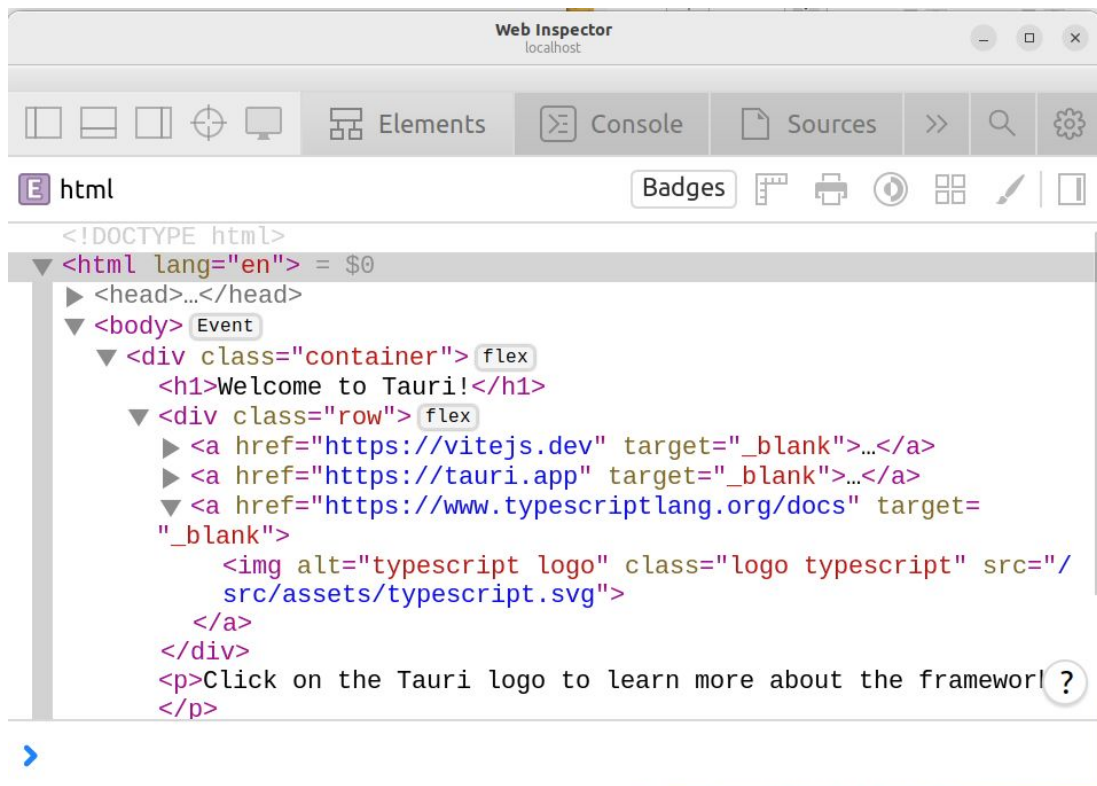
21ms

Enhances the experience of Ta...

Tauri



Débogage - JS



Commande dev

➔ **airsoft-training** npm run tauri dev

\$ tauri dev

Running BeforeDevCommand (`npm run start`)

\$ ng serve

Application bundle generation complete. [2.147 seconds] (watching)

➔ **Local:** <http://localhost:1420/>

Info Watching src-tauri for changes...

Compiling airsoft-training v0.0

Finished dev [unoptimized + debuginfo] target(s) in 11.34s

Commande build

➔ **airsoft-training** npm run tauri build

```
$ tauri build
```

```
    Running beforeBuildCommand `npm run build`
```

```
$ ng build
```

```
Application bundle generation complete. [10.479 seconds]
```

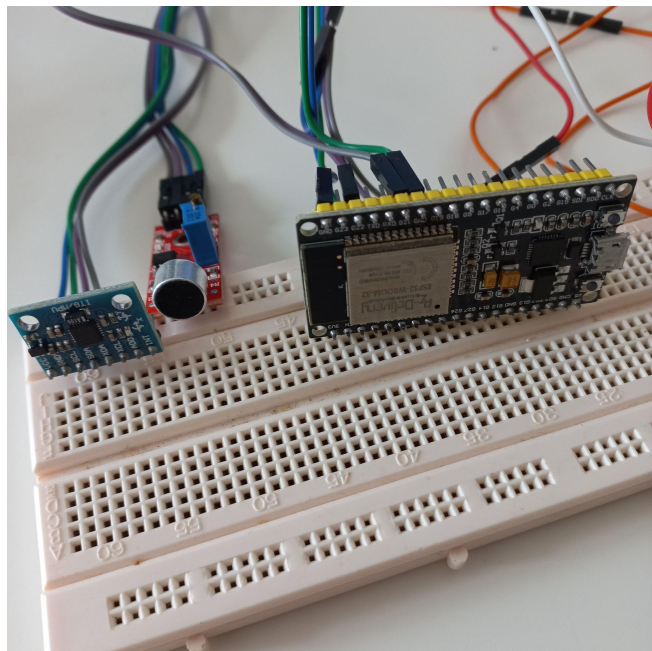
```
    Compiling airsoft-training v0.0.0
```

```
    Finished release [optimized] target(s) in 18.58s
```

```
    Bundling airsoft-training_0.0.0_amd64.AppImage
```

Formats de sorties configurables

Prototype prêt !



Connection OK

CONNEXION

DÉCONNEXION

Nombre tirs :

3

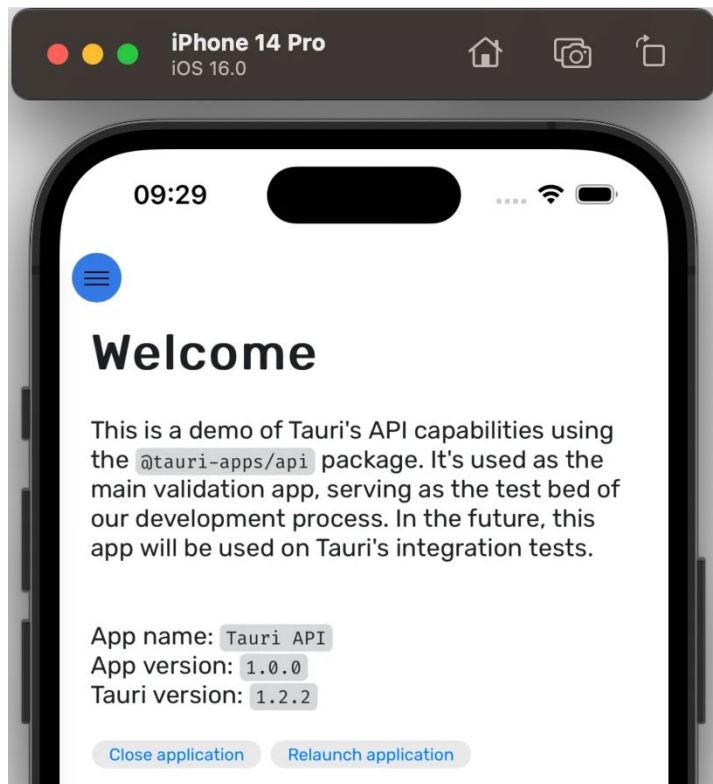
DÉMARRER

Tir #	Temps accéléré (ms)	Temps micro (ms)	Différence (ms)
-------	---------------------	------------------	-----------------

1	-	-	-
---	---	---	---

Application Angular pour tests

Futur



Alors, Tauri ou Electron ?

Prêt à monter en
compétences sur
Rust ?

Utile dans d'autres
contextes, ex:
WebAssembly,
Oxlint (rust eslint)

Libs Rust
disponible pour les
fonctionnalités
attendues ?

Besoin de
fonctionnalités
webview
spécifiques ?

Support
Windows < 10

Webview non intégré

Taille codebase à
migrer ? (si
existante)

Merci !

Avez-vous des questions ?

