

Démarrer un projet Electron

Electron permet de transformer une application Web en application native. Plusieurs frameworks JS (Angular, Vue.js, React, Svelte, etc) peuvent fournir l'app Web de départ et trois plateformes peuvent être choisies comme cibles : win32 (Windows), linux (Linux), et darwin (macOS).

Dans ce tutoriel nous créerons une app **Angular** et la transformerons en un .exe pour Windows

Prérequis :

1- On suppose qu'**Angular** est installé sur votre poste de travail. Si ce n'est pas le cas voici une procédure pour l'installer :

https://amartel.techinfo-cstj.ca/4203D4/01-angular_installation.pdf

2- Toutes les commandes du tutoriel se font à l'**Invite de Commande de Windows (ICW)**. On suppose une connaissance minimale de cet environnement.

3- On suppose que vous ayez un compte sur **github.com**. Si ce n'est pas le cas, créez-en un :

<https://github.com/>

Prêt? On commence :

- 1- Créez un nouveau projet Angular appelé **poker24** et installez-y Bootstrap
- 2- Dans un éditeur de code, modifiez le template html du component de base (**app.component.html**) pour qu'il affiche simplement un message d'accueil :
`<h1>Bienvenue dans Poker24</h1>`
- 3- Dans une ICW positionnez-vous à la racine du projet et lancez le serveur Web de développement Angular :

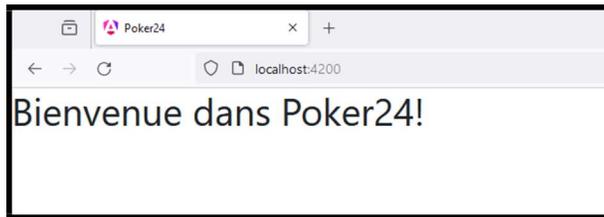
ng serve

```
C:\atelier\5C8\Aut24\poker24>ng serve
Initial chunk files | Names | Raw size
styles.css | styles | 277.63 kB |
polyfills.js | polyfills | 88.34 kB |
main.js | main | 1.53 kB |
| Initial total | 367.50 kB

Application bundle generation complete. [3.186 seconds]

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
→ Local: http://localhost:4200/
→ press h + enter to show help
```

- 4- Ouvrez votre navigateur favori et entrez l'URL du serveur Web de développement Angular (localhost :4200). Vous devriez voir ceci :



- 5- Nous avons la base d'un projet Angular, nous allons la faire évoluer vers un projet **Electron** :

Dans le dossier ...\poker24\src\ modifiez le fichier **index.html** :
modifiez `<base href="/">` pour `<base href=".">`

- 6- Dans l'ICW, arrêtez le serveur (Ctrl-C). Vérifiez que vous êtes sur le bon dossier (...\poker24\ : le dossier contenant le fichier **package.json**) et faire

```
npm install --save-dev electron
```

```
C:\atelier\5C8\Aut24\poker24>npm install --save-dev electron
added 52 packages, and audited 1002 packages in 1m

150 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
```

Note : Le paramètre « --save-dev » signifie que le package **Electron** sera installé dans la section devDependencies de package.json. Ceci permet de garder les dépendances de développement séparées des dépendances de production, ce qui allégera l'installation pour les utilisateurs finaux qui n'ont pas besoin des dépendances de développement. Les applications seront livrées avec le binaire Electron, ce qui élimine ainsi le besoin de le spécifier comme dépendance de production.

- 7- Toujours à la racine du projet (...\poker24\) créez le fichier **main.js**. Ce fichier est le point d'entrée de toute application Electron. **Mettez-y** le code suivant:

```

const {app, BrowserWindow} = require('electron');
const url = require('url');
const path = require('path');

function onReady () {
  win = new BrowserWindow({width: 300, height: 670 })
  win.loadURL(url.format({
    pathname: path.join(
      __dirname,
      'dist/poker24/browser/index.html'),
    protocol: 'file:',
    slashes: true
  )))
}
app.on('ready', onReady);

```

Attention : vérifiez minutieusement que le path fourni est identique au path existant sur votre disque. S'il y a une erreur minuscule/majuscule votre app .exe donnera une page blanche

8- Dans **package.json** veuillez ajouter les éléments suivants :

```

{
  "main": "main.js",
  "author": "votre nom",
  "description": "app pour jouer au poker",
  "scripts": {
    "electron": "electron .",

```

9- Pour rouler un des scripts définis dans package.json il suffit de taper :

npm run nomDuScript

10- La commande «**electron .**» permet d'exécuter l'app en mode développement. Pour l'exécuter on tape :

npm run electron

Si on la roule maintenant nous frapperons une erreur :

```

C:\atelier\5C8\Aut24\poker24>npm run electron
> poker24@0.0.1 electron
> electron .

(node:38592) electron: Failed to load URL: file:///C:/atelier/5C8/Aut24/poker24/dist/poker24/browser/index.html with error: ERR_FILE_NOT_FOUND
(Use `electron --trace-warnings ...` to show where the warning was created)

```

11- L'erreur se produit parce que l'app Angular n'a pas encore été buildée, donc le fichier ...\\poker24\\dist\\poker24\\browser\\index.html est inexistant. Donc avant d'appeler **electron .**, nous devons lancer un build. Ajoutons une commande :

```

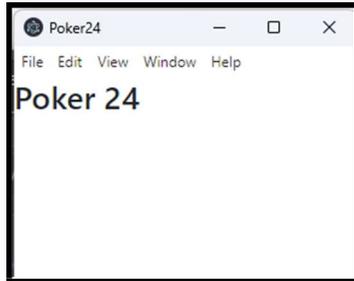
"electron-build": "ng build && electron .",
description: "app pour jouer au poker",
  "scripts": {
    "electron": "electron .",
    "electron-build": "ng build && electron .",
    "ng": "ng",
    "start": "ng serve"

```

12- Lançons maintenant le script **electron-build** :

```
npm run electron-build
```

L'app, sous forme d'un exe windows, devrait s'exécuter !!



13- Pour enlever la barre de titre et le menu par défaut on peut modifier le main.js ainsi :

```
. . .  
function onReady () {  
  win = new BrowserWindow({width: 300, height: 670, frame: false, })  
. . .
```

14- Puis relancer **npm run electron** (pas besoin de ng build parce qu'on a seulement modifié le main.js, fichier qui concerne Electron et qui est ignoré par Angular)

15- Votre app est rudimentaire, mais nous avons une structure de base pour générer une application native Windows (.exe) à partir d'une app Angular.
Nous verrons plus loin comment la builder et la livrer mais avant, créons un repo github pour conserver et distribuer notre application.

Git et Angular

Lors de la création d'un projet Angular (`ng new NomProjet`) le CLI d'Angular fournit un dossier `.git` contenant un `.gitignore` adapté à un projet Angular. Nous nous servons de ces outils pour conserver et faire évoluer notre code source.

1- « Gitter » votre projet :

- Positionnez-vous sur le dossier racine de votre projet (... \poker24\)
- **git status**
- **git add .**
- **git commit -m"initial"**
- Allez sur **github.com**
- Créez un repository git nommé **poker24** (private, pas de readme, pas de gitignore)
- Copiez l'URL de ce repository
- Retour sur l'env angular
- **git remote add origin https://github.com/utilisateurGit/Poker24.git**
- **git push origin master**

2- Le projet est maintenant disponible sur GitHub. Si vous voulez l'installer sur un autre ordi :

- Positionnez-vous sur le dossier où vous désirez installer le projet
- **git clone https://github.com/utilisateurGit/Poker24.git**
- **cd Poker24**
- **npm install** (installera les packages Bootstrap et Electron)
- À ce stade vous serez synchro avec github

Builder et distribuer une app Electron

Une fois votre application **Electron** fonctionnelle en mode dév, il faut utiliser des outils supplémentaires pour créer l’empaquetage et le livrable. À la base Electron n’inclut pas ces outils. Nous utiliserons **Electron Forge** qui est un outil spécialisé dans l’empaquetage et la distribution des applications Electron.

- 1- Dans une ICW positionnez-vous à la racine du projet Electron (...\\poker24\\)
- 2- Installez les outils de **electron-sourceforge** en tapant :
npm install --save-dev @electron-forge/cli
- 3- Lancez la commande:
npx electron-forge import
Note : Cette commande modifie la config du package.json en ajoutant, entre autres, les scripts **"make"** et **"package"**.
- 4- Regardez dans la section « scripts » de package.json les commandes disponibles:

```
package.json
{
  "name": "Poker24",
  "version": "0.0.1",
  "main": "main.js",
  "author": "votre disgracieux nom",
  "description": "créateur et gestionnaire de parties de poker",
  "scripts": {
    "ng": "ng",
    "start": "electron-forge start",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test",
    "electron": "electron .",
    "electron-build": "ng build && electron .",
    "package": "electron-forge package",
    "make": "electron-forge make"
  },
}
```

- 5- Générez votre application :
npm run make
Cette commande est longue (environ 2 min. sur ma machine), elle créera le dossier ...\\Poker24**out**\\ où l'on retrouvera deux sous-dossiers :
make et **Poker24-win32-x64**
- 6- Le dossier **make** contient le fichier distribuable :
...\\Poker24\\out\\make\\squirrel.windows\\x64**Poker24-0.0.1 Setup.exe**
C'est un programme d'installation de notre app. Il peut être copié et exécuté sur une autre machine windows, il exécutera notre app.

Le dossier **Poker24-win32-x64** contient, entre autres, un fichier **Poker24.exe** qui peut être lancé directement sur la machine locale de développement.

- 7- Copiez le fichier **Poker24-0.0.1 Setup.exe** (distribuable) sur une machine Windows étrangère à notre environnement de développement et lancez ce programme. Vous pouvez maintenant développer une app native avec les outils du Web! C'est l'heure de se réjouir et d'être épaté!