
Fiche technique découverte d' Electron.js

Qu'est une application de bureau ?

Un appli de bureau est un logiciel installer sur notre ordi (ex: GotoMeeting) tandis qu'un appli web est un session qui a lieu entièrement depuis une fenêtre de navigation

Qu'est qu'électron ?

Electron est une librairie open source développé par GitHub pour créer des applications bureau multi-platform (mac Windows linux) développer avec des techno web HTML, CSS et JavaScript

Exemple de logiciel de bureau : VSC Twitch sont développer avec électron

Avantages :

Simple d'installations

Utilisation des technos web

Avoir ou crée des composants (notification, alertes, boutons ect..)

Coder en React Angular et Vue est possible

Inconvénients :

Pour un petit projet électrons occupé beaucoup de place 50 et 150Mo (intègre un navigateur web)

Autre techno pour crée des appli web :

Lorca

Webview

Prérequis pour utiliser électron

VCS, Node js, Git

Sources :

<https://maori.geek.nz/golang-desktop-app-webview-vs-lorca-vs-electron-a5e6b2869391>

<https://www.electronjs.org/docs/tutorial/quick-start#package-and-distribute-the-application>

<https://www.youtube.com/watch?v=bmFbqgxJcf8>

<https://fr.jeffprod.com/blog/2017/developpement-d-applications-multiplateforme-avec-electron/>

```
D:\documents\Test_l-tron>npm install -g electron
C:\Users\sautr\AppData\Roaming\npm\electron -> C:\Users\sautr\AppData\Roaming\npm\node_modules\electron\cli.js

> core-js@3.6.5 postinstall C:\Users\sautr\AppData\Roaming\npm\node_modules\electron\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)
```

```
D:\documents\Test_l-tron>git clone https://github.com/electron/electron-quick-start
Cloning into 'electron-quick-start'...
remote: Enumerating objects: 693, done.
Receiving objects: 92% (638/693), 236.01 KiB | 418.00 KiB/s 693 receiving objects: 90% (624/693), 236.01 KiB | 418.00 KiB/s
Receiving objects: 100% (693/693), 453.62 KiB | 508.00 KiB/s, done.
Resolving deltas: 100% (369/369), done.
```

```
D:\documents\Test_l-tron\electron-quick-start>npm install

> core-js@3.6.5 postinstall D:\documents\Test_l-tron\electron-quick-start\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> electron@10.1.2 postinstall D:\documents\Test_l-tron\electron-quick-start\node_modules\electron
> node install.js

added 87 packages from 97 contributors and audited 87 packages in 7.742s

6 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Installez les dépendances et lancez l'app

```
$ npm install -g electron
```

Clonez le dépôt Quick Start

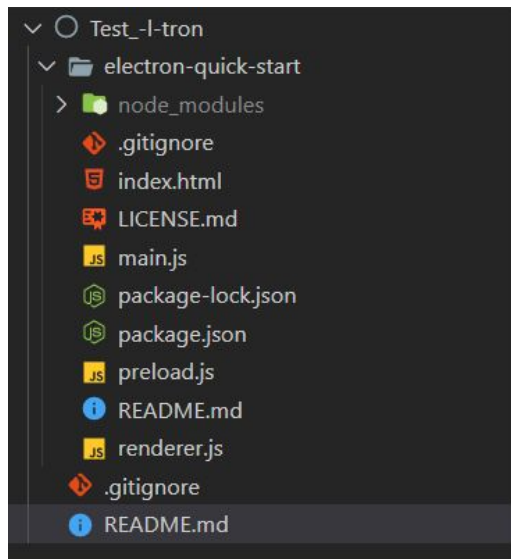
```
$ git clone https://github.com/electron/electron-quick-start
```

Allez dans le dépôt

```
$ cd electron-quick-start
```

Installez les dépendances et lancez l'app

```
$ npm install && npm start
```



Hello World!

We are using Node.js 12.16.3, Chromium 85.0.4183.98, and Electron 10.1.2.

AUTRE MÉTHODE AVEC PACKAGE NPM

-Crée son repo

-Initialiser son repo avec npm init

-Installer le package `npm i -D electron@latest`

-Dans votre repo crée un fichier app.js a la racine de votre projet comme ceci

```
const path      = require('path');

const express    = require('express');

const expressApp = express();

const http       = require('http').Server(expressApp);


// Import des models

/* Mettre les relation ici */

// Routes handler

const index = require('./routes/router');

/* variable initialisation's */

const router = {

  isStarted: false

};

/**

 * Starting web server on port 3000

 *

 * When we start we create tables in database if not exist

 * @param {*} callback

 */

function start(callback) {

  if (router.isStarted === false) {
```

```

init(function () {

    // Handle routes function

    loadRoutes(function () {

        // starting web server

        http.listen(3000, function () {

            console.log('Application is running on port 3000');

            router.isStarted = true;

            if (typeof callback != 'undefined') {

                callback();

            }

        });

    });

});

} else {

    console.log("Application already started");

    if (typeof callback != 'undefined') {

        callback();

    }

}

}

/**
 * Initialisation of view engine and others parameters
 * @param {*} callback
 */
function init(callback) {

```

```

/** view engine setup*/
expressApp.set('views', path.join(__dirname, 'views'));
expressApp.set('view engine', 'ejs');
expressApp.use(express.json());
expressApp.use(express.urlencoded({ extended: false }));
expressApp.use(express.static(path.join(__dirname, 'public')));

/* Keep server down */
router.isStarted = false;
if (typeof callback !== 'undefined') {
    callback();
}
}

/**
 * Route's management
 * @param {*} callback
 */
function loadRoutes(callback){
    expressApp.use("/", index);
    if (typeof callback !== 'undefined') {
        callback();
    }
}

module.exports = {

```

```
start: start  
};
```

-Crée tjr à la racine un fichier main.js c'est avec le fichier qui permet l'ouverture de l'app desktop, le fichier main.js doit être liée à votre fichier serveur app.js ou server.js

```
const electron = require('electron');  
  
const app = require('./app');  
  
let window;  
  
function createWindow() {  
  /* Créer une fenêtre de 800px par 600px sans bordures */  
  window = new electron.BrowserWindow({  
    width: 800,  
    height: 700,  
    frame: false  
  });  
  
  /* Si vous décommentez cette ligne, vous verrez la console de debug Chrome */  
  // window.webContents.openDevTools();  
  
  /* Display the homepage of the server */  
  window.loadURL('http://127.0.0.1:3000');  
  
  /* Lorsque la fenêtre est fermée, on l'indique au système */  
  window.on('closed', () => {  
    window = null;  
  });  
}
```

```

}

/* On attend qu'Electron.js soit prêt pour créer la fenêtre */
electron.app.on('ready', function () {

    app.start(function () {

        createWindow();

    });

});

/* Cette fonction arrête totalement l'application
   lorsque toutes les fenêtres sont fermées */
electron.app.on('window-all-closed', () => {

    if (process.platform !== 'darwin') {

        electron.app.quit();

    }

});

/* Fonction utile pour MacOS */
electron.app.on('activate', () => {

    if (win === null) {

        createWindow();

    }

});

```

-Pour relancer automatique l'app après modif comme nodemon on peut installer electronmon via `npm i electronmon`

-Dans le package.json indiquer que vous lancez l'app avec le fichier main.js, si on utilise electronmon utiliser la syntaxe suivante, **FAIT UN npm run start-electron ou electronmon main.js pour lancer l'app**


```
{
  "name": "",
  "version": "1.0.0",
  "description": "",
  "main": "main.js",
  "scripts": {
    "start-electron": "electron ./main.js",
    "start": "node ./app.js"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/Oceane97440/calculatrice.git"
  },
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/Oceane97440/calculatrice/issues"
  },
  "homepage": "https://github.com/Oceane97440/calculatrice#readme",
  "dependencies": {
    "ejs": "^3.1.5",
    "electromon": "^1.0.10",
    "express": "^4.17.1",
  },
  "devDependencies": {
    "electron": "^10.1.3"
  }
}
```

