

Création d'une application native (MacOS, Windows, Linux)

Installation de Node.js

Installation de Node.js :

- Rendez-vous sur le site officiel de Node.js à l'adresse suivante <https://nodejs.org>.
- Téléchargez la version recommandée pour votre système d'exploitation.
- Exécutez le programme d'installation téléchargé et suivez les instructions pour installer Node.js.
- Vous pouvez vérifier que Node est bien installé en tapant la commande suivante dans votre terminal `node -v`. Ce dernier vous renverra la version de node installée sur votre ordinateur.
- Si vous possédez déjà node, vérifiez que vous possédez bien une des dernières versions.

Création des fichiers json et js

Dans un premier temps vous allez devoir créer un dossier répertoire dans lequel tous les fichiers seront positionnés.

Dans votre terminal positionnez vous à l'intérieur de ce dossier. Pour se faire il faut utiliser la commande `cd` suivi du chemin vers votre répertoire. Par exemple si votre fichier se nomme AppTest et réside dans le bureau alors la commande à effectuer sera: `cd Desktop/apptest`.

Si vous avez bien exécuter la commande alors le terminal se positionnera dans votre répertoire.

```
Last login: Mon Jun 26 15:35:51 on ttys000
ingefox@Ingefoxs-Mac-mini ~ % cd ~/Desktop/apptest
ingefox@Ingefoxs-Mac-mini apptest %
```

Dans le terminal positionné dans votre répertoire vous allez créer le fichier `package.json` avec la commande suivante: `npm init -y`.

Ensuite vous allez ouvrir un éditeur de code et l'ouvrir avec votre dossier répertoire.

Vous allez y créer un fichier `index.js`. Attention si vous décidez de modifier le nom de ce fichier il faudra également le faire dans le fichier json à l'endroit correspondant.

Ensuite vous allez inclure ce code dans votre fichier `index.js`:

```
const { app, BrowserWindow } = require('electron');
const { url } = require('inspector');
const path = require('path');
const util = require('util');
const donnees = require('./package.json')
const mainUrl = donnees.url;

function createWindow() {
  // Create the browser window.
  const mainWindow = new BrowserWindow({
    show: false
  });

  // Gestion des cookies lorsqu'ils sont modifiés
  let cookies = mainWindow.webContents.session.cookies;
  cookies.on('changed', function(event, cookie, cause, removed) {
    if (cookie.session && !removed) {
      let url = util.format('%s://%s%s', (!cookie.httpOnly && cookie.secure) ? 'https' :
'http', cookie.domain, cookie.path);
      cookies.set({
        url: url,
        name: cookie.name,
        value: cookie.value,
        domain: cookie.domain,
        path: cookie.path,
        secure: cookie.secure,
        httpOnly: cookie.httpOnly,
```

```

        expirationDate: new Date().setDate(new Date().getDate() + 14)
    }, function(err) {
        if (err) {
            console.error('Erreur lors de la persistance du cookie', err, cookie);
        }
    });
}
});

mainWindow.loadURL(mainUrl); // Remplacez l'URL par celle que vous souhaitez charger

// Événement déclenché lorsque la fenêtre principale est prête à être affichée
mainWindow.once('ready-to-show', () => {
    // Retire le menu en haut de page
    mainWindow.removeMenu();
    // Maximise la fenêtre
    mainWindow.maximize();
    // Affiche la fenêtre
    mainWindow.show();
});
}

// Événement déclenché lorsque Electron a fini de s'initialiser
app.on('ready', createWindow);

```

En récupérant l'objet `cookies`, on peut gérer les cookies de manière dynamique, notamment en réagissant aux changements qui se produisent, tels que l'ajout ou la modification d'un cookie. Cela permet d'effectuer des opérations telles que la persistance des cookies, en les sauvegardant avec une durée d'expiration spécifiée, ou d'effectuer d'autres traitements en fonction des cookies présents.

Vous pouvez ajouter des modifications à votre code JavaScript suivant vos besoins, par exemple l'ouverture maximisée n'est pas obligatoire vous pouvez la changer.

Construction de l'application

Ici vous allez voir comment générer des fichiers exécutables pour les 3 systèmes d'exploitations. (Mac Os, Windows, Linux)

Première étape: Installation d'electron

Vous allez installer électron localement dans votre fichier répertoire avec la commande `npm install electron --save-dev`.

Deuxième étape: Installation d'électron builder

Pour installer electron builder vous allez exécuter la commande suivante: `npm install electron-builder --save-dev`.

Troisième étape: Ajout de script dans le fichier package.json

Vous allez ajoutez ces 2 scripts dans le fichier package.json.

```
"url": "https://votre-url.com",           // Permet de ne pas stocker l'url dans le fichier JS

"scripts": {
  "build:mac": "electron-builder --mac",
  "build:win": "electron-builder --win",
  "build:linux": "electron-builder --linux"
}

=====

"build": {
  "appId": "com.example.my-electron-app",
  "mac": {
    "category": "your.app.category.type",
    "target": "dmg",
    "icon": "/Chemin/vers/mon/icone.icns"    // A remplacer
  },
  "win": {
    "target": "nsis",
    "icon": "/Chemin/vers/mon/icone.ico"    // A remplacer
  },
  "linux": {
    "target": "AppImage",
    "icon": "/Chemin/vers/mon/icone.png"    // A remplacer
  }
}
```

Dernière étape: Création de l'application et des fichiers executables

Pour générer maintenant les fichiers exécutables vous allez taper la commande suivante:

```
npm run build: <votreOS>
```

 . Pour l'OS ce sera soit "win" soit "mac" ou encore "linux".

A l'intérieur de votre dossier répertoire se trouvera un dossier "dist" contenant un fichier setup pour l'installation de votre application. Suivez ensuite les instructions d'installation.

Révision #16

Créé Fri, Jun 23, 2023 11:35 AM par **Thibault**

Mis à jour Tue, Jul 4, 2023 7:50 AM par **Stage**