**Introduction du logiciel Electron**

Le logiciel Electron est un environnement qui permet de développer des applications multi-plateformes de bureau avec JavaScript, HTML et CSS. Il a été créé en 2013. L’infrastructure est codée en Node.js et l’interface est bâtie sur les outils Chromimum (partie open source de Google Chrome). C’est un logiciel libre open source développé par Git Hub. Les logiciels et applications célèbres qui ont été développé à l’aide d’Electron sont les éditeurs de texte libres Atom, Visual Studio Code et l’application Discord. La dernière version est 32.0.0 qui est sortie le 14 octobre 2024.

**Fonctionnement du logiciel**

Les navigateurs web exercent de nombreuses responsabilités secondaires, telles que la gestion de plusieurs fenêtres (ou onglets) et le chargement d’extensions tierces. Avant, les navigateurs utilisaient généralement un processus unique pour toutes ces fonctionnalités. C’est moins de frais, mais quand un site Web plantait ou se bloquait, cela pouvait affecter l’ensemble du navigateur. Pour résoudre ce problème, Chrome a décidé que chaque onglet serait rendu dans son propre processus. Cela limite les dommages que le code bogué ou malveillant d’une page Web pourrait causer à l’application. Un seul processus de navigateur contrôle alors ces processus, ainsi que le cycle de vie de l’application dans son ensemble. Les applications Electron sont structurées de manière très similaire. En tant que développeur on contrôle deux types de processus. Le **«** MainProcess **»** (processus de principal) contrôlele cycle de vie del’application et le **«** Renderer Process **»** (processus de rendu distinct)gèrel’interface utilisateur.

**Processus principal**

L’objectif du processus principal est de créer et de gérer les fenêtres avec le module **«** BrowserWindow **»**. Chaque instance de la classe **«** BrowserWindow **»** crée une fenêtre d’application qui charge une page Web dans un processus de rendu distinct. On peut interagir avec ce contenu Web à partir du processus principal a l’aide de l’objet **«** webContents ». Le processus contrôle également le cycle de vie de l’application avec le module **«** app ». Le **«** MainProcess **»** ajoute également des API pour interagir avec les systèmes d’exploitation de l’utilisateur. Electron a de plus, plusieurs modules qui contrôlent les fonctionnalités natives du bureau, comme les menus, les boite de dialogue et les icones de la barre d’état.

**Processus de rendu distinct**

Le processus de rendu distinct est responsable du rendu du contenu web. Le code exécuté dans les processus de rendu doit se comporter conformément aux normes web. Les interfaces utilisateur et fonctionnalités d’application au sein d’une seule fenêtre de navigateur doivent être écrites avec les mêmes outils qu’on utilise sur le web. Certaines spécification web sont : Un fichier HTML est le point d’entrée pour le processus de rendu, le style de l’interface utilisateur est ajouté via des fichiers CSS et du code JavaScript exécutable peut être ajouté. Il n’y a pas de moyen d’importer les scripts de contenu d’Electron, donc on utilise les scripts de préchargement. Ils contiennent du code qui s’exécute dans un processus de rendu avant que son contenu Web ne commence à se charger. Ils s’exécutent dans le contexte du moteur de rendu. Ils ont accès aux API Node.js. Le script sert à améliorer le moteur de rendu. Ils exposent des API arbitraires et le contenu Web peut ensuite les consommer.

**Installation**

Pour l’installation, on doit installer Node.js. Il faut créer un dossier et initialiser un package npm. Ensuite, pour exécuter Electron, il faut activer la commande npm start. Electron s’exécuter à la racine du projet. Lors de l’exécution, Electron va chercher script **«** main». C’est ce script qui contrôle le processus principal.Il faut donc initialiser le script **«** main». Il faut créer un fichier vide nommé main.js dans le dossier racine du projet.

**Avantages et inconvénients**

Ce logiciel utilise les langages pour la programmation web. Ce qui est plus facile pour les personnes qui maitrises seulement HTML, CSS et JavaScript. Un autre avantage, c’est que ce Framework peut être utilisé dans plusieurs plateformes. Les applications peuvent fonctionner dans Windows, Linux et Mac. Pour les inconvénients, Les applications qu’on crées sont lourdes à cause de Chromimum et Node. Elles peuvent contenir beaucoup de package et de dépendances en plus. Une application Electron peut dépasser les 200-300 Mo. Elles prennent de plus beaucoup de mémoire.

**Sécurité Dans Electron**

Les risques de sécurité sont généralement faibles. Cependant, les risques de sécurité augmentent avec les pouvoirs supplémentaires accordés au code. C’est important de mettre l’application à jour. Les vulnérabilités critiques sont normalement corrigées quand on met à jour le Framework vers la dernière version. Il faut aussi faire attention aux bibliothèques qu’on installe.

**Annexe : veille technologique**

1. **Professionnel humain**

**Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement**

**Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement**

**Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement**

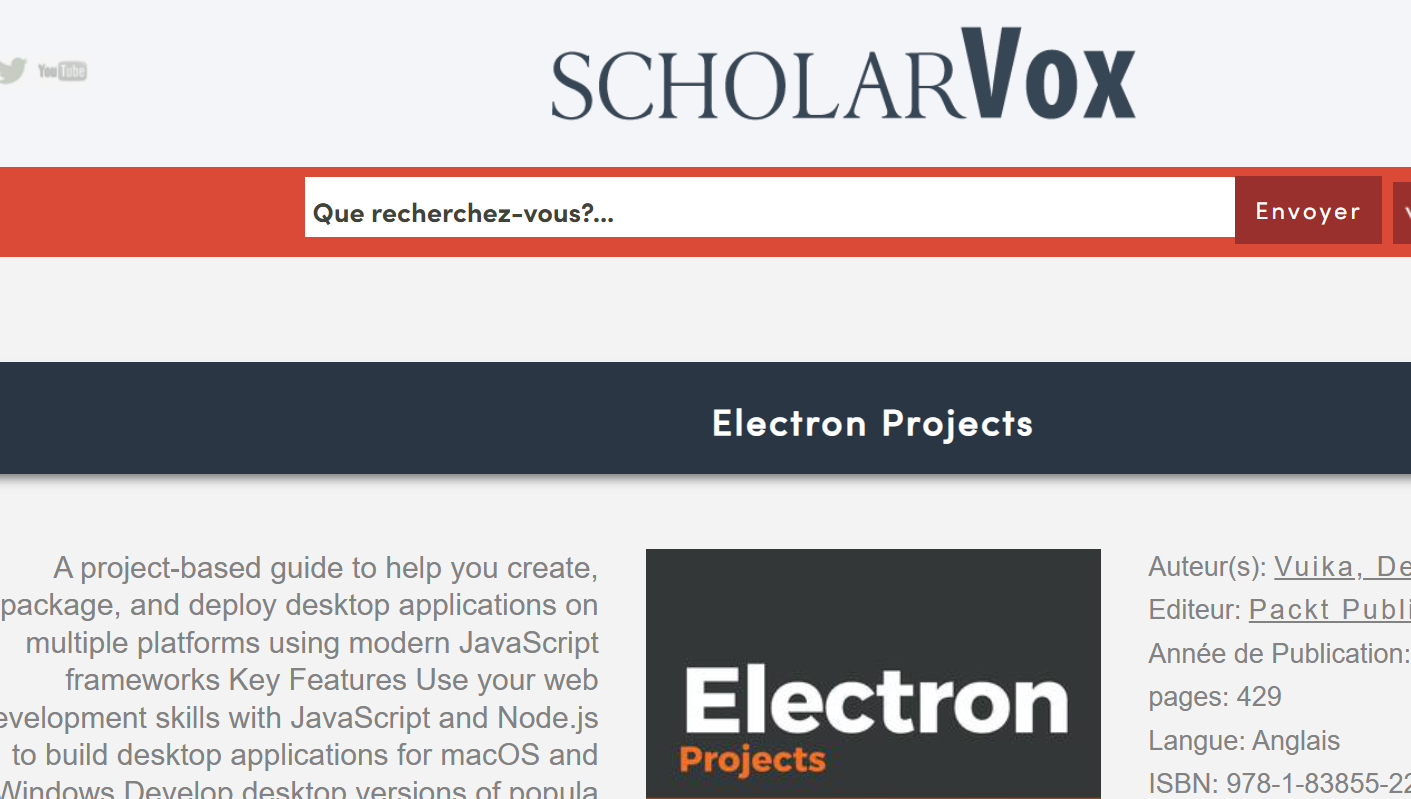
**Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement**

**J’ai fait un poste sur une communauté reddit Electronjs et des experts m’ont répondu (je n’ai pas demandée leur métier).**

1. **Moteurs de recherche spécialisés**

**ScholarVox :**

****

**Je l’ai trouvé dans la section catalogue bibliothèque de l’école et j’ai trouvé un livre utile pour ce projet.**

**Semantic scholar : Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement**

**J’ai utilisé Semantic Scholar pour trouver des articles et d’autres informations pertinentes.**

**WorldCat :**

**Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement**

**J’ai aussi essayé WorldCat, mais je n’ai trouvé grand-chose.**

**Google Scholar :**

**Une image contenant texte, capture d’écran, Police, Page web

Description générée automatiquement**

**J’ai aussi utilisé Google Scholar pour d’autres ouvrages académiques.**

1. **Suivi de mots-clés avec Google Alertes**

**Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement**

**Les mots clés utilisés sont Electron updates, Cross-platform apps Electron, Electron Node.js, Electron security et Electron framework pour avoir les mises à jour et nouveautés de ce logiciel.**

1. **Flux RSS**

**Une image contenant texte, logiciel, Page web, Icône d’ordinateur

Description générée automatiquement**

**J’ai utilisé Freedly pour les Flux RSS avec Electron js, Smashing magazine et Dev.to. Ce sont des blogs qui parlent d’Electron et de technologie en général.**

**Une image contenant texte, logiciel, Page web, Icône d’ordinateur

Description générée automatiquement**

**J’ai aussi utilisé Inoreader, car je n’étais pas sur si Feedly marchait. Je me suis abonnée a d’autres blog de technologie.**

1. **Médias sociaux**

**Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement**

**Je me suis abonnée sur twitter au compte officiel Electron.**

**Une image contenant texte, capture d’écran, logiciel, multimédia

Description générée automatiquement**

**C’est le compte de Node.js. Surveiller des nouveautés sur Node.js.**

**Une image contenant texte, capture d’écran, logiciel, multimédia

Description générée automatiquement**

**Frontend Masters : c’est un compte de cours et d’informations crées par des experts**

1. **Surveillance de pages Web**

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

**Node.js Blog**

Une image contenant texte, logiciel, Page web, Site web

Description générée automatiquement

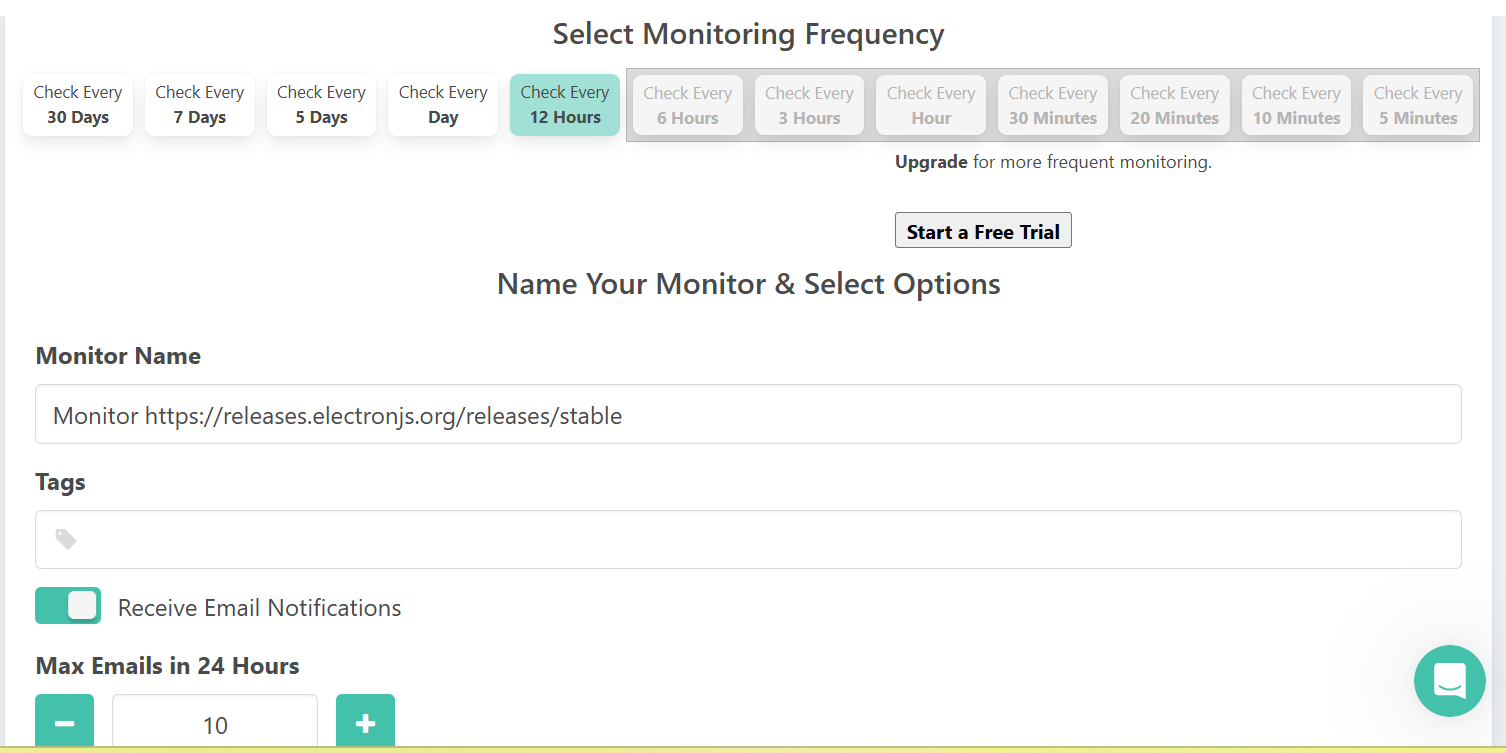
Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

**Electron Release Notes**

**Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement**

****

**GitHub Electron Issues**

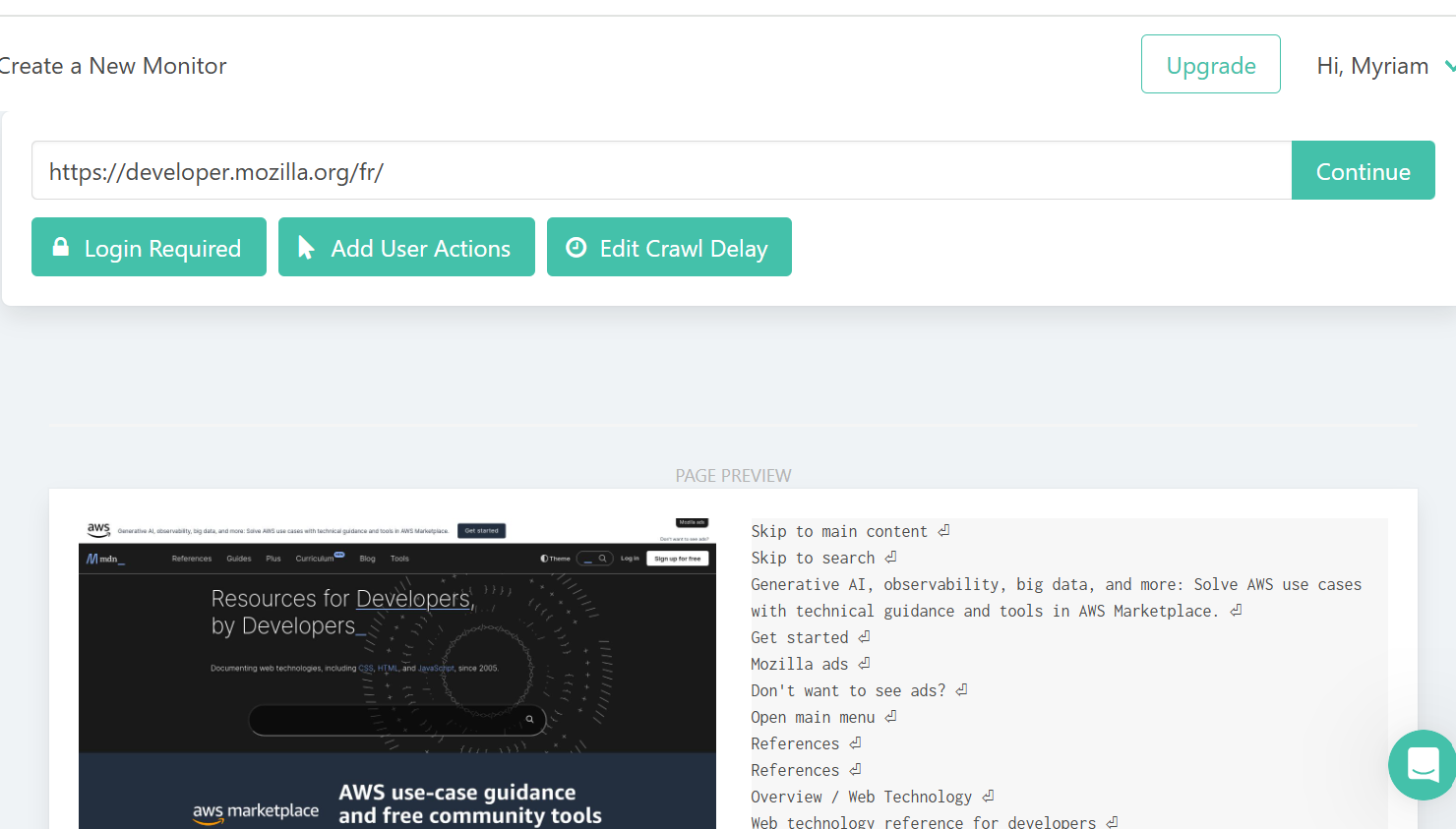
**Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement**

**Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement**

**Tutorials Electron**

****

**Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement**

**MDN Web Docs**

**Bibliographie**

<https://www.electronjs.org/docs/latest>

<https://fr.wikipedia.org/wiki/Electron_(framework)>

<https://talks.freelancerepublik.com/electron-javascript-creer-apps-desktop/#:~:text=Comme%20nous%20l'avons%20vu,soutenu%20par%20une%20grosse%20communaut%C3%A9>

<https://cyberlibris-montmorency.proxy.collecto.ca/catalog/book/88876732>