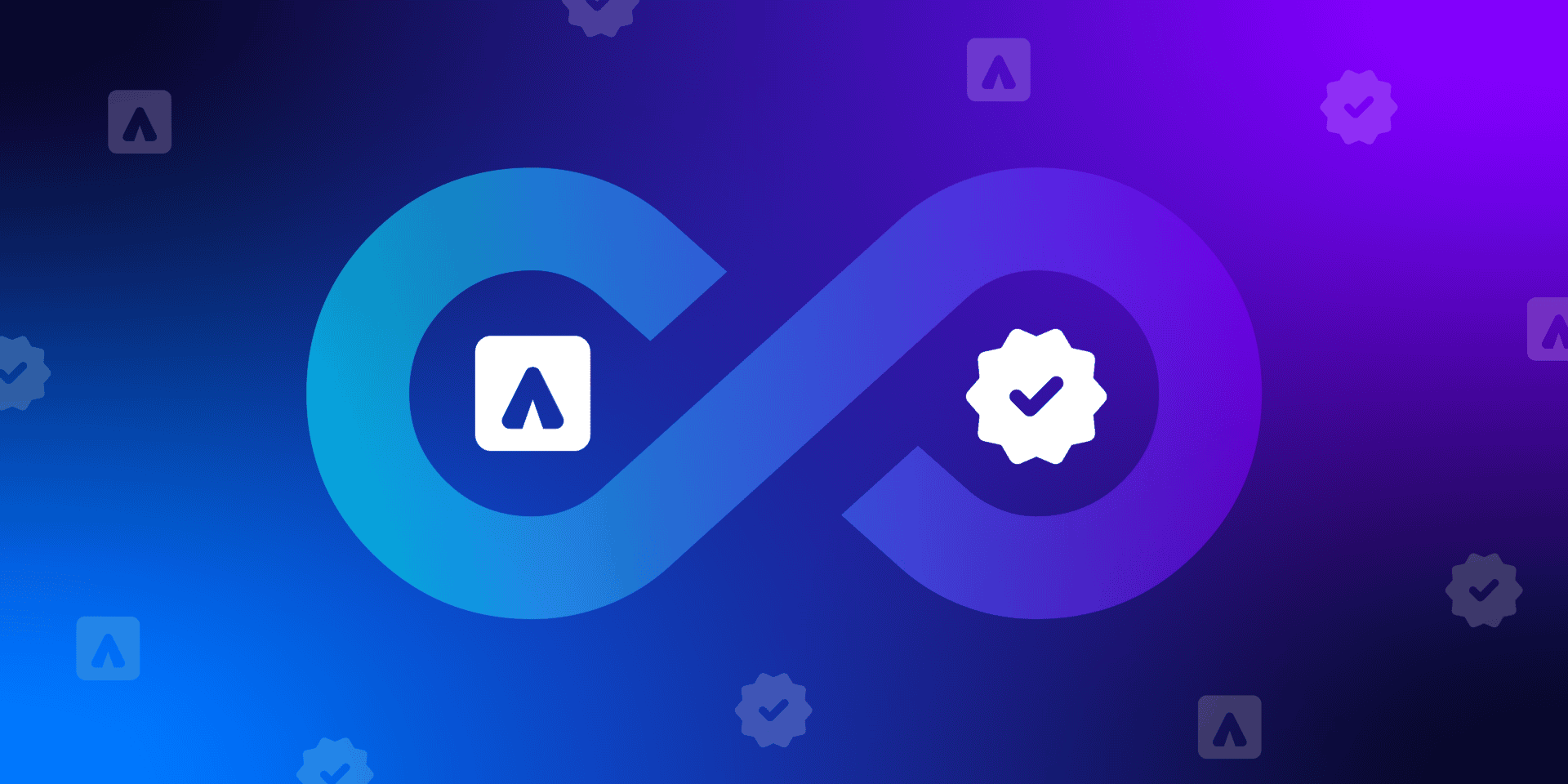
**PORTFOLIO**



William Trelles – Classe CID 3A

ETML-Lausanne-Vennes

Professeur : Mathieu Meylan

Table de matière

[1 Introduction 2](#_Toc187574814)

[2 Analyse et planification initiale 2](#_Toc187574815)

[3 Configuration de l’environnement de développement 2](#_Toc187574816)

[4 Développement et tests 3](#_Toc187574817)

[5 Planification / jobs / étapes CI/CD 3](#_Toc187574818)

[6 Résultats finaux et conclusion 4](#_Toc187574819)

# Introduction

Le projet "addon-todo-app" est une extension développée pour le navigateur Firefox permettant aux utilisateurs de gérer leurs tâches de manière efficace. Ce portfolio présente les étapes de conception, de développement et de déploiement de cette application avec une approche CI/CD.

# Analyse et planification initiale

L'objectif principal de cette extension était de permettre :

La gestion des tâches personnelles sous forme de liste.

Un déploiement automatisé pour simplifier la maintenance et les mises à jour.

Pour atteindre cet objectif, plusieurs étapes ont été définies :

Mise en place d’un environnement de développement compatible avec les outils CI/CD.

Configuration des workflows GitHub Actions pour automatiser les processus de build et de déploiement.

La planification est accessible à travers ce [lien](https://github.com/users/witrellesp/projects/2/views/1).

# Configuration de l’environnement de développement

L'environnement de développement a été configuré avec les outils suivants :

* Node.js (v20) : Utilisé pour la gestion des dépendances et des scripts.
* wxt : Un outil de construction pour les extensions de navigateur.
* GitHub Actions : Configuration des workflows CI/CD pour automatiser les étapes de build, test et déploiement.
* Secrets GitHub : Stockage des clés sensibles et variables nécessaires au déploiement sécurisé de l'extension.

Procédure de mise en place :

Installation de Node.js.

Initialisation du projet avec les dépendances via npm ci.

Configuration du fichier wxt.config.ts pour inclure les variables dynamiques comme ADD\_ON\_FIREFOX\_VERSION et FIREFOX\_ADDON\_ID.

Configuration des secrets et des variables globales dans GitHub.

# Développement et tests

# Planification / jobs / étapes CI/CD

## Étape 1 : Checkout

Le workflow utilise actions/checkout@v3 pour cloner le dépôt sur la machine virtuelle. Cela garantit que tout le code source est disponible pour les étapes suivantes.

## Étape 2 : Configuration de Node.js

La version 20 de Node.js est installée pour exécuter les commandes npm. C'est une version récente et stable compatible avec les outils utilisés.

## Étape 3 : Installation des dépendances

La commande npm ci installe toutes les dépendances nécessaires depuis package-lock.json. Cette méthode garantit que les versions exactes des dépendances sont utilisées.

## Étape 4 : Export des variables

Les variables ADD\_ON\_FIREFOX\_VERSION (version de l’add-on) et FIREFOX\_ADDON\_ID (UUID de l’add-on) sont exportées dans l'environnement. Ces valeurs sont utilisées par les étapes suivantes, notamment pour configurer le fichier manifest.json.

## Étape 5 : Audit des vulnérabilités

npm audit fix détecte et corrige automatiquement les vulnérabilités de sécurité dans les dépendances.

## Étape 6 : Build

La commande npm run build:firefox génère les fichiers nécessaires à l’add-on dans le répertoire de sortie (.output/firefox).

## Étape 7 : Compression

La commande npm run zip:firefox compresse l’add-on en un fichier .zip prêt à être déployé. Le fichier compressé est généré dans .output/firefox/.

## Étape 8 : Déploiement

La commande utilise l’action wdzeng/firefox-addon@v1 pour uploader le fichier .zip sur le Firefox Add-ons Developer Hub.

Les secrets FIREFOX\_JWT\_ISSUER et FIREFOX\_JWT\_SECRET sont utilisés pour s’authentifier auprès de l’API de déploiement.

# Résultats finaux et conclusion

L’extension a été compilée et déployée avec succès. Voici les étapes clés obtenues :

Build et test locaux : L’extension a été testée localement avec npm run build:firefox et npm run zip:firefox.

Déploiement automatique : Le workflow CI/CD a permis d’automatiser le déploiement vers Firefox Add-ons avec un fichier .zip correct tous les fichiers nécessaires.

