

# Exercice 2 - Automatiser lint, artefacts et versioning avec GitHub Actions

---

Dans cet exercice, vous allez mettre en place un mini-pipeline CI/CD simple autour d'un petit projet NodeJS, incluant linting, génération d'artefacts et gestion automatique des versions via des tags. Cet exercice s'inscrit dans les objectifs du module 324 – Prendre en charge des processus DevOps.

## Objectifs de l'exercice

- Créer un mini-projet NodeJS.
- Mettre en place un workflow CI qui exécute un lint et génère un artefact téléchargeable.
- Automatiser la gestion du versionnage (tags Git) avec un second workflow.
- Manipuler des déclencheurs GitHub Actions.
- Utiliser une action externe depuis GitHub Marketplace.

## Étapes à réaliser

### 1. Créer un mini projet NodeJS

- Exemple minimal : un petit serveur web avec `http.createServer()` ou Express.
- Initialiser le projet

### 2. Configurer ESLint

- Installer ESLint
- Initialiser ESLint
- Vérifier que `npm run lint` fonctionne

### 3. Créer un repo GitHub

- Créer un repository privé
- Partagez votre repository avec votre enseignant
- Ajouter votre projet, puis effectuer un push

### 4. Workflow CI sur la branche main

Sur la branche `main`, créer un workflow GitHub Actions

- se déclenche au push sur `main`
- installe NodeJS
- exécute le lint
- crée un artefact contenant votre projet

Le workflow doit s'exécuter sans erreur et permettre le téléchargement de l'artefact depuis l'interface GitHub.

### 5. Créer un tag `v1.0.0`

En local :

```
git tag v1.0.0  
git push origin v1.0.0
```

## 6. Créer un workflow manuel pour générer un tag automatiquement

Créer un second workflow qui :

- est déclenché manuellement
- propose un menu déroulant avec trois choix : **major**, **minor**, **patch**
- utilise une action GitHub Marketplace pour incrémenter automatiquement le numéro de version selon le choix
- crée un **nouveau tag Git** en conséquence

PS

Pour chercher une action adaptée :

<https://github.com/marketplace?type=actions>