

# Auto Git Commit - Guide d'installation

Ce script Node.js automatise les commits Git sur GitHub avec une interface web moderne utilisant Tailwind CSS.

## Fonctionnalités

- **Commits automatiques** : 1-2 commits toutes les heures avec variation aléatoire
- **Interface web moderne** : Dashboard avec Tailwind CSS et Alpine.js
- **Messages personnalisables** : Liste de messages de commit configurable
- **Monitoring en temps réel** : Statistiques et statut en direct
- **Commits manuels** : Possibilité de déclencher des commits à la demande
- **Sécurité** : Pas de stockage de mots de passe, utilise l'authentification Git locale

## Prérequis

1. **Node.js** (version 14 ou supérieure)
2. **Git** configuré avec authentification GitHub
3. **Repository Git** initialisé et connecté à GitHub

## Installation

### 1. Créer le projet

```
bash

# Créer un nouveau dossier
mkdir auto-git-commit
cd auto-git-commit

# Initialiser le projet npm
npm init -y
```

### 2. Installer les dépendances

```
bash

# Installer les dépendances principales
npm install express node-cron

# Installer les dépendances de développement (optionnel)
npm install --save-dev nodemon
```

### 3. Créer les fichiers

Créez un fichier `server.js` avec le code du script principal fourni.

### 4. Configuration Git

Assurez-vous que Git est configuré avec vos identifiants :

```
bash

# Configurer Git (si pas déjà fait)
git config --global user.name "Votre Nom"
git config --global user.email "votre.email@example.com"

# Pour GitHub, configurez l'authentification par token ou SSH
# Exemple avec token (recommandé) :
git remote set-url origin https://YOUR_TOKEN@github.com/username/repository.git
```

### 5. Initialiser le repository (si nécessaire)

```
bash

# Si c'est un nouveau repository
git init
git add .
git commit -m "Initial commit"
git branch -M main
git remote add origin https://github.com/username/repository.git
git push -u origin main
```



## Utilisation

### Démarrer le serveur

```
bash

# Mode production
npm start

# Mode développement (avec auto-reload)
npm run dev
```

### Accéder à l'interface

Ouvrez votre navigateur et allez sur : `http://localhost:3000`







## Interface utilisateur

## Dashboard principal

- **Statut en temps réel** : Affiche si l'automation est active
- **Statistiques** : Nombre total de commits, dernier commit
- **Contrôles** : Boutons pour démarrer/arrêter l'automation

## Fonctionnalités disponibles

1.  **Démarrer/Arrêter** : Active ou désactive l'automation
2.  **Commit manuel** : Déclenche un commit immédiatement
3.  **Actualiser** : Met à jour les statistiques
4.  **Configuration** : Modifier le chemin du repo et les messages

## Configuration











Dans la section configuration, vous pouvez :

- **Chemin du repository** : Spécifier le dossier Git à utiliser
- **Messages de commit** : Personnaliser la liste des messages aléatoires

## Configuration avancée

### Messages de commit par défaut

Le script utilise ces messages par défaut :

-  Update documentation
-  Fix minor issues
-  Add new features
-  Bug fixes
-  Improve code structure
-  Performance improvements
-  Security updates
-  Mobile responsiveness
-  Internationalization
-  Deploy updates

## Fréquence des commits

- **Fréquence** : Toutes les heures
- **Variation** : Délai aléatoire de 0-30 minutes
- **Nombre** : 1 ou 2 commits par exécution

## Types de changements générés

Le script crée automatiquement :

1. **README.md** : Mise à jour avec timestamp
2. **auto-commit.log** : Fichier de log des commits
3. **version.json** : Fichier de version avec build number

## Sécurité et bonnes pratiques

### Authentification GitHub

#### Recommandé : Personal Access Token

```
bash
```

```
# Créer un token sur GitHub (Settings > Developer settings > Personal access tokens)
```

```
# Puis configurer Le remote :
```

```
git remote set-url origin https://TOKEN@github.com/username/repo.git
```

#### Alternative : SSH

```
bash
```

```
# Utiliser une clé SSH configurée
```

```
git remote set-url origin git@github.com:username/repo.git
```

### Considérations importantes

- **Repository privé recommandé** : Pour éviter le spam public
- **Pas de données sensibles** : Le script ne génère que des fichiers de test
- **Respecter les politiques GitHub** : Utiliser avec modération
- **Backup** : Garder une copie de votre code important ailleurs

## Dépannage

### Erreurs communes

### 1. "Permission denied"

- Vérifiez l'authentification Git
- Assurez-vous que le token a les bonnes permissions

### 2. "Not a git repository"




- Vérifiez que vous êtes dans un dossier Git initialisé
- Configurez le bon chemin dans l'interface

### 3. "Failed to push"

- Vérifiez la connexion internet
- Assurez-vous que le repository existe sur GitHub

## Logs de débogage

Le script affiche des logs dans la console :

-  Succès des commits
-  Erreurs avec détails
-  Informations de programmation

## Personnalisation

### Modifier la fréquence

Dans `server.js`, ligne avec `cron.schedule('0 * * * *', ...)` :

- `'0 * * * *'` = toutes les heures
- `'*/30 * * * *'` = toutes les 30 minutes
- `'0 */2 * * *'` = toutes les 2 heures

### Ajouter des types de fichiers

Dans la fonction `createDummyChanges()`, ajoutez vos propres générateurs de contenu.

### Personnaliser l'interface

L'interface utilise Tailwind CSS et Alpine.js, vous pouvez la modifier dans la route `app.get('/', ...)`.

## Licence

MIT License - Libre d'utilisation et de modification.

## Avertissement

Utilisez ce script de manière responsable. Les commits automatiques peuvent être détectés par GitHub si utilisés de manière excessive. Recommandé pour des projets personnels ou de test uniquement.