

Prérequis

- [Node.js](#) (v16+)
- Git installé
- Un compte GitHub
- VS Code (recommandé) avec l'extension ESLint

1 Initialisation du projet et installation d'ESLint

1. Créez un nouveau projet et initialisez Git :

```
mkdir tp-eslint-git && cd tp-eslint-git
```

```
git init
```

```
npm init -y
```

2. Installez ESLint :

```
npm install eslint --save-dev
```

3. Configurez ESLint :

```
npx eslint --init
```

4. Créez un `.gitignore` :

```
echo "node_modules/" >> .gitignore
```

```
echo ".eslintcache" >> .gitignore
```

2 Test d'ESLint sur un fichier JavaScript

1. Créez un fichier `app.js` avec des erreurs :

```
const x=10;  
  
console.log(x);  
  
function test(){  
  
  console.log('test')  
  
}  
  
test()
```

2. Lancez ESLint :

```
npx eslint app.js
```

- Observez les erreurs (indentation, `console.log`, etc.).

3. Corrigez le fichier manuellement ou avec `--fix` :

```
npx eslint --fix app.js
```

3 Intégration avec Git Hooks (Husky)

1. Installez Husky (pour les hooks Git) :

```
npm install husky --save-dev
```

```
npx husky install
```

2. Ajoutez un hook `pre-commit` pour lancer ESLint :

```
npx husky add .husky/pre-commit "npx eslint ."
```

3. Testez le hook :

```
git add .
```

```
git commit -m "Test du hook ESLint"
```

- Si le code contient des erreurs, le commit est bloqué.

4 Configuration avancée d'ESLint

1. Modifiez `.eslintrc.json` pour personnaliser les règles :

```
{  
  "extends": "airbnb-base",  
  "rules": {  
    "no-console": "warn",  
    "indent": ["error", 2],  
    "quotes": ["error", "single"]  
  },  
  "env": {  
    "browser": true,  
    "node": true  
  }  
}
```

2. Ajoutez un script `lint` dans `package.json` :

```
"scripts": {  
  "lint": "eslint ."  
}
```

3. Testez la configuration :

```
npm run lint
```

5 Mise en place de GitHub Actions

1. Créez un dépôt GitHub et liez-le
2. Ajoutez un workflow CI (`./github/workflows/lint.yml`) :

```
name: Lint Code
```

```
on: [push, pull_request]
```

```
jobs:
```

```
  lint:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

- uses: actions/checkout@v3
- uses: actions/setup-node@v3
- with:- node-version: 18
- run: npm ci
- run: npm run lint

3. Poussez les changements et vérifiez l'exécution sur GitHub :

```
git add .
```

```
git commit -m "Ajout de GitHub Actions"
```

```
git push
```

- Allez dans l'onglet Actions de votre dépôt pour voir le résultat.

6 Simulation d'un travail d'équipe

1. Créez une branche avec du code non conforme :

```
git checkout -b feature/ajout-fonction
```

- Ajoutez un fichier avec des erreurs ESLint (ex : `utils.js`).
2. Essayez de commiter (le hook doit bloquer) :

```
git add .
```

```
git commit -m "Code non conforme"
```

3. Corrigez les erreurs et créez une Pull Request :

```
npx eslint --fix .
```

```
git add .
```

```
git commit -m "Correction des erreurs ESLint"
```

```
git push --set-upstream origin feature/ajout-fonction
```

- Vérifiez que GitHub Actions valide la PR.
- Soumettez vos repositories GIT et vos résultats d'erreur dans le readme