**Linting & Formatage dans un projet Node.js / Express.js**

* **ESLint**

**ESLint** est un outil de **linting** pour JavaScript.  
Son objectif est de **détecter automatiquement** les erreurs potentielles, les mauvaises pratiques, ou les incohérences dans le code.  
Il analyse le code **statistiquement** (sans l'exécuter) et peut signaler :

* Des erreurs de syntaxe,
* Des variables non utilisées,
* Des fonctions mal définies,
* Des règles non respectées (comme l’ordre des imports ou le style des accolades).
* **Prettier**

**Prettier** est un **formateur automatique de code**.  
Son rôle est de **réécrire ton code** selon un style défini (indentation, guillemets, espaces, retour à la ligne, etc.) **sans modifier sa logique**.  
Il ne se concentre pas sur les erreurs, mais sur l’**uniformité visuelle**.

Exemple :

function test() {

function test( ) {console.log("salut")} 👉🏾 console.log("salut");

}

* **Comment ESLint et Prettier se complètent**
* **ESLint** vérifie que ton code **respecte les bonnes pratiques** et **évite les erreurs**.
* **Prettier** rend ton code **plus lisible et cohérent**.
* Utilisés ensemble, ils assurent un code **propre, lisible, sans erreur, et uniforme**.
* **Pourquoi sont-ils importants en équipe ?**

Dans un projet collaboratif, chaque développeur peut avoir son propre style de code. Sans convention stricte, le code devient vite chaotique, difficile à relire et source de nombreux conflits dans Git. En utilisant des outils comme ESLint et Prettier, toute l’équipe est alignée sur les mêmes règles – quels que soient l’éditeur ou les habitudes de chacun.

**Résultat :**  
Une base de code propre, cohérente et professionnelle, avec moins de conflits, et un gain de temps précieux à long terme.