

## 5- BENCHMARK D'OUTILS

Outils	Langages	Avantages	Inconvénients
Artillery	YAML / JSON / JS	<ul style="list-style-type: none"><li>- Très utilisé dans l'écosystème Node.js</li><li>- Scénarios écrits en YAML ou JSON</li><li>- Simule des utilisateurs virtuels avec différents parcours</li><li>- Bon pour simuler du trafic sur des APIs REST ou WebSockets</li><li>- Génère des rapports clairs en JSON ou HTML</li></ul>	<ul style="list-style-type: none"><li>- Moins rapide que des outils très bas-niveau</li><li>- Moins adapté aux charges extrêmes comparé à k6</li></ul>
k6	JavaScript (ES6), développé en Go	<ul style="list-style-type: none"><li>- Ultra performant, supporte des milliers de VU</li><li>- Tests API complexes</li><li>- Léger, idéal pour CI/CD</li><li>- Scripts en JS modernes (ES6)</li><li>- Reporting visuel via k6 Cloud ou HTML reports</li></ul>	<ul style="list-style-type: none"><li>- Interface cloud payante si on veut des fonctionnalités avancées</li><li>- Moins simple à prendre en main qu'Artillery</li></ul>
Autocannon	CLI ou Node.js	<ul style="list-style-type: none"><li>- Très simple pour du benchmark brut</li><li>- Ultra rapide</li><li>- Affiche latences et requêtes/seconde</li></ul>	<ul style="list-style-type: none"><li>- Pas de scénarios complexes</li><li>- Pas de reporting visuel</li></ul>
Jest + Puppeteer	JavaScript	<ul style="list-style-type: none"><li>- Permet de tester l'UI et l'API ensemble</li><li>- Parfait pour l'e2e (End to End)</li><li>- Simule l'utilisateur réel (navigateur)</li></ul>	<ul style="list-style-type: none"><li>- Pas conçu pour tester de fortes charges</li><li>- Peu performant à grande échelle (utilise un navigateur réel)</li></ul>

**Tableau illustrant un Benchmark d'outils JS par rapport à Locusts**