

Proposer un contenu

Identifiant

Identifiant

Mot de passe

Mot de passe

☐ Connexion automatique

Se connecter

Pas de compte ? S'inscrire...

# Journal : Je vous présente mon projet entièrement généré par IA

Posté par n\_e le 17 octobre 2026 à 19:55.

Étiquettes : slop



## Sommaire

- [Les outils](#)
- [Comment j'ai procédé](#)
- [Mes impressions](#)
  - [L'extension Codex pour VSCode est fonctionnelle, mais pas terrible](#)
  - [C'est rapide](#)
  - [L'IA n'est pas un dev senior, et n'aide pas à progresser](#)
  - [Peu de tokens sont utilisés](#)
- [Ma conclusion générale](#)

J'avais envie de tester l'IA sur un projet un peu plus complexe que ce que j'ai fait jusque-là, à savoir essentiellement des scripts et utilitaires à usage unique ou presque.

Pour cela, j'ai "écrit" un outil de recherche pour LinuxFR, à l'image de celui qui était disponible il y a quelques années via la zone de saisie "recherche" en haut à droite.

L'outil est accessible ici : <https://linuxfr-search.ztel.fr> (il est en train d'indexer, donc en fonction de quand vous lisez le journal vous pouvez ne voir que 1000 contenus) et le code là : <https://github.com/n-e/dlfp-search>.

## Les outils

J'ai choisi l'offre ChatGPT Plus à 20€/mois, qui était réputée plus généreuse en tokens que l'offre équivalente Anthropic, et avait un mois gratuit. Coté client, j'ai utilisé l'extension Codex pour VSCode, et coté modèle GPT-6 Luna en niveau d'effort "moyen".

## Comment j'ai procédé

J'avais une idée assez précise de l'architecture et du résultat que je souhaitais, par contre sur les détails, j'ai préféré laisser l'IA faire.

Pour cela, j'ai écrit une [mini-spec](#), j'ai demandé à l'IA de me faire un retour et me poser des questions, suite à quoi elle a mis à jour [les specs](#).

Ensuite, je lui ai demandé de coder les parties indexeur, backend et frontend. Par exemple, pour le backend, je lui ai simplement demandé "write the backend" dans un nouveau chat, ce qui a donné [ce résultat](#).

Ensuite je lui ai demandé des ajustements. Pour les bugs, je l'ai laissé trouver par lui-même avec exemple la requête "run the backend and send a request" à laquelle il a répondu :

```
The backend is running on port 3000. /health returned 200 OK. After fixing the Elasticsearch _score sort , /search?q=linux&pageSize=1 returned 200 OK with one result and pagination showing 3 total matches. I also fixed a Node.js TypeScript runtime issue that prevented the server from starting.
```

Ça fonctionne aussi pour les bugs où la solution est plus subjective. Par exemple je lui ai demandé "get the index and check for problems" car les pages étaient mal parsées et il m'a répondu :

```
The linuxfr alias resolves and searches successfully. It contains 10 documents, all indexed from the sample as news; a full-text search for “logiciel libre” returned relevant records. I found a few quality limitations in the sample: - Extracted content includes page noise, such as a leading score/date and “Aller plus loin” links with click counts. - All 10 authors are Anonyme. That may be accurate for these older posts, but the sample doesn't confirm author extraction for named authors. - Only 4 records have tags, and the sample contains no journals or forum posts, so those types remain unverified. - title is mapped as a text field without a keyword subfield, so Elasticsearch won't support sorting or aggregations on it directly. A diagnostic aggregation I tried failed for this reason; it doesn't affect full-text search.
```

Enfin, pour les modifications, j'ai demandé ce que je voulais, par exemple "now simplify the frontend design. I don't want mono fonts or silly taglines. Also this project is not endorsed by linuxfr so have the title something like "linuxfr.org search", not linuxfr".

## Mes impressions

### L'extension Codex pour VSCode est fonctionnelle, mais pas terrible

J'ai rencontré 3 soucis avec l'extension :

- elle est très lourde (plus de 1go), ce qui paraît énorme par rapport aux fonctionnalités offertes
- elle est lente. Par exemple afficher un diff prend environ 1s, alors que l'outil de diff intégré de VSCode (et n'importe quel autre outil de diff) est instantané.
- une des raisons pourquoi j'ai choisi Codex est qu'il exécute l'agent dans un bac à sable. Ça n'a pas été le cas par défaut (alors que c'était le cas pour la version CLI), et j'ai ensuite eu pas mal de problèmes de permissions (par exemple des requêtes réseau qui ne passaient pas alors que je pensais les avoir autorisées).

Pour le dernier point, les problèmes sont largement dus à mon manque d'expérience, mais j'en ai quand même retiré l'impression que la fonctionnalité de sandbox n'était pas utilisée par la majorité des développeurs, alors que ça me semble pourtant être la base.

### C'est rapide

Il m'a fallu environ 3 heures pour coder le projet, itérations et déploiement inclus. Sans IA je pense qu'il m'aurait fallu une journée, mais avec un résultat de meilleure qualité.

Cela étant dit, quand on pointe un problème à l'IA elle est efficace pour le corriger, donc j'aurais pu prendre le temps libéré pour améliorer les choses.

### L'IA n'est pas un dev senior, et n'aide pas à progresser

Pour le premier point, l'IA fait beaucoup de choix de pertinence moyenne. Par exemple dans le projet, le choix des librairies JavaScript (utilisation de deux librairies de parsing XML plutôt qu'utiliser JSDom) ou des mappings Elasticsearch n'est pas le meilleur. Elle n'utilise pas non plus la façon la plus robuste de parser les pages (pas les bons sélecteurs CSS, parsing de chaînes de caractères plutôt qu'utiliser des éléments dédiés, etc).

Pour le 2e point, j'ai été surpris de voir que l'IA ne propose pas des alternatives, et ne fait pas de retour sur les choix demandés. Au contraire si on lui demande quelque chose de stupide, ou qui parait stupide avec les éléments à sa disposition, elle le fait sans broncher.

Par contre, quand on lui demande de corriger quelque chose ça fonctionne bien.

Ma conclusion, c'est que si on connaît déjà la bonne manière de faire, l'IA est très efficace, par contre pour quelqu'un de plus junior, il va voir du code avec des mauvaises pratiques et sans doute, sans retours d'une autre personne, arriver aux mauvaises conclusions.

### Peu de tokens sont utilisés

Le projet a utilisé 5% de ma limite sur 5h, et 1% de ma limite hebdomadaire sur l'offre ChatGPT+ à 20€/mois, alors que ce sont les offres plus chères qui sont recommandées pour le code. La prochaine fois je testerai avec un modèle plus performant et plus cher.

## Ma conclusion générale

Au final c'était une expérience positive. Par rapport au codage "manuel", c'est un peu plus rapide, et surtout on n'est pas concentré sur le code, ce qui permet de réfléchir aux idées de plus haut niveau (architecture, le changement suivant à faire, etc.) pendant que l'IA code, ce qui est très agréable.

La qualité de code par défaut est moyenne, mais, du moins pour un projet de cette taille, l'IA est efficace pour faire les corrections. Par exemple, le frontend par défaut avait un design très slop, et quand je lui ai demandé de corriger elle est arrivée à quelque chose de beaucoup plus sobre, et a supprimé le CSS qui n'était plus pertinent. Je suis également certain qu'elle aurait été efficace pour faire d'autres améliorations que je n'ai pas pris la peine de faire (par exemple remplacer les fonctions qu'elle a codé à la main par lodash ou autre).

Ma dernière conclusion c'est que, contrairement à ce qu'on lit habituellement, c'est que ça me paraît être un outil adapté aux développeurs expérimentés. Si on ne la guide pas avec une bonne architecture, on ne corrige pas rapidement les mauvaises directions dans laquelle elle part, on peut vite arriver à quelque chose de non fonctionnel, non performant et non maintenable. En plus, lorsqu'on code à la main on est souvent obligés de décider (par ex. est-ce que je code ça à la main ou je trouve une librairie qui le fait ?), alors que l'IA fait un choix par défaut qui n'est pas toujours le bon, et il faut savoir le remettre en cause.

J'encourage tout le monde à tester sur un petit projet et se faire son propre avis.

(0 commentaire). [Markdown](#) [EPUB](#)

Envoyer un commentaire

Suivre le flux des commentaires

**Note** : les commentaires appartiennent à celles et ceux qui les ont postés. Nous n'en sommes pas responsables.

[Revenir en haut de page](#)

#### Derniers commentaires

- Re: Ah ? Ça ira
- Re: Ah ? Ça ira
- Re: Euro-office desk...
- Re: Euro-Office est i...
- Re: Calcul économiq...
- Euro-office desktop
- Re: Libera Suite est-...
- Re: Euro-Office est i...
- Libera Suite est-elle ...
- Re: C'est un peu triste
- Re: inatteignables
- Euro-Office est illégal

#### Étiquettes (tags) populaires

- intelligence\_artificielle
- slop
- merdification
- états-unis
- grands\_modèles\_de...
- capitalisme
- bigtech
- administration\_fran...
- politique
- python
- spam
- startup\_nation

#### Sites amis

- Agenda du Libre
- April
- Éditions D-BookeR
- Éditions Diamond
- Éditions Eyrolles
- Éditions ENI
- En Vente Libre
- Framasoft
- La Quadrature du Net
- Lea-Linux
- Open Source Initiative
- Imprimerie Grafik Plus

#### À propos de LinuxFr.org

- Mentions légales
- Faire un don
- L'équipe de LinuxFr...
- Informations sur le s...
- Aide / Foire aux que...
- Suivi des suggestion...
- Wiki du site
- Règles de modération
- Statistiques
- API pour le dévelop...
- Code source du site
- Plan du site