

Code Buddy 2

Un agent qui code dans votre dépôt, sous vos règles d'approbation, sans facture d'API supplémentaire.

Code Buddy lit votre dépôt, écrit du code, exécute des commandes et rend compte — dans le terminal, dans une application de bureau ou derrière une API HTTP. Il s'appuie sur un abonnement que vous payez déjà ou sur un modèle local, et reste sous votre contrôle : chaque écriture et chaque commande passent par la règle d'approbation que vous avez fixée.

État : Version 2.0.0 publiée le 9 septembre 2026 sur npm, paquet @phuetz/code-buddy. Le dépôt compte 2 182 fichiers de tests automatisés ; seules les vérifications d'intégration continue sous Linux sont bloquantes, macOS et Windows tournent au mieux.

À qui ça sert

Les équipes qui veulent maîtriser la dépense

Un abonnement existant ou un modèle local suffit à faire tourner l'agent : le coût marginal peut rester nul, avec un plafond de dépense par session.

Ceux qui ne peuvent pas envoyer leur code dehors

Le mode entièrement local, avec Ollama ou LM Studio, fait tourner l'agent sans qu'une ligne de code quitte la machine.

Les développeurs qui veulent une boucle vérifiée

L'agent ne s'arrête pas quand il se croit fini, mais quand la commande de vérification que vous avez donnée se termine correctement.

Ce que ça change concrètement

1 Une boucle qui s'arrête sur une preuve, pas sur une promesse

La commande buddy loop planifie, modifie, lance votre commande de vérification et ne se termine que si celle-ci réussit. Un correctif annoncé mais non prouvé ne passe pas.

2 Un routeur unique devant 64 fournisseurs de modèles

Claude, GPT, Grok, Gemini, Ollama, LM Studio, NVIDIA NIM et les autres passent par la même interface, avec bascule automatique quand un fournisseur sature. Une trentaine de chemins gratuits ou locaux sont recensés.

3 Plus de 220 outils, choisis selon la question

Lecture, édition, exécution, recherche, navigation : les outils utiles sont sélectionnés à chaque requête plutôt que tous déclarés, ce qui économise le contexte.

4 Une flotte, pas un agent isolé

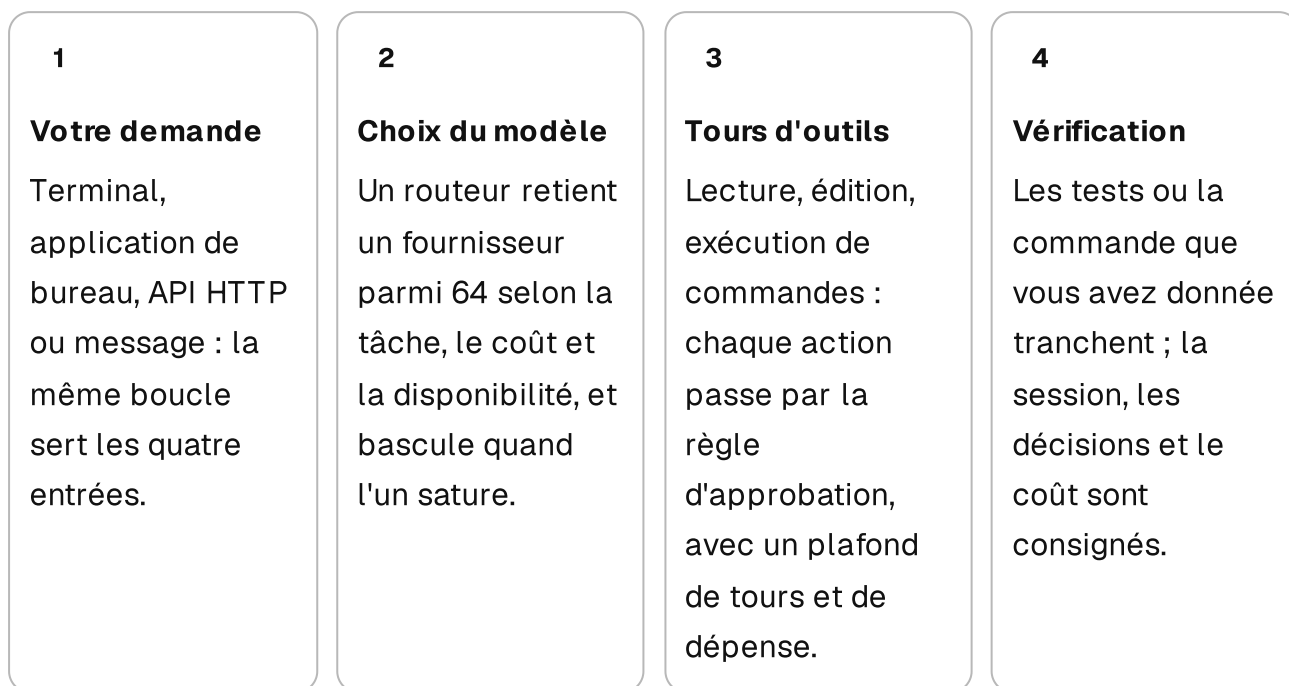
Plusieurs instances lancées en serveur s'observent et s'appellent : conversation entre pairs, sessions multi-tours, exécution d'outils distants en lecture seule derrière trois verrous successifs.

5 Une application de bureau et une surface HTTP

Cowork apporte l'exécution visuelle de flux, la bibliothèque média et le studio vidéo. Une API compatible OpenAI permet de brancher vos propres clients.

Comment ça marche

L'agent tourne en boucle : il choisit un outil, l'exécute sous votre règle d'approbation, lit le résultat, et recommence jusqu'à la vérification.



Les fonctions avancées — écriture spéculative, mémoire collective, auto-amélioration, perception — sont toutes désactivées par défaut : sans leur variable d'environnement, le comportement reste identique.

Chiffres vérifiables

Chaque chiffre ci-dessous est daté et provient du dépôt ou d'une mesure publiée. Rien n'est extrapolé.

64 fournisseurs
modèles accessibles derrière un routeur unique, dont une trentaine de chemins gratuits ou locaux

Source : README et description du paquet npm. Un recomptage interne du 22 août 2026 relève 65

220 outils et plus
outils disponibles, sélectionnés requête par requête

Source : README du dépôt.

entrées de catalogue pour 64 identifiants uniques.

2 182 fichiers de tests

fichiers de tests automatisés dans le dépôt — des fichiers, pas des cas de test

Source : Comptage direct dans le dépôt le 12 septembre 2026. Le dépôt refuse désormais d'afficher un total de cas de test qui n'aurait pas été remesuré.

2.0.0

version du paquet npm

@phuetz/code-buddy, publiée le 9 septembre 2026

Source : Registre npm public, champ de date de publication.

Pour démarrer

Trois commandes suffisent, avec Node.js 20 ou plus récent. L'authentification par abonnement ChatGPT évite toute clé d'API.

1. Installer et se connecter

```
npm i -g @phuetz/code-buddy
buddy login      # abonnement ChatGPT : aucune clé d'API
buddy           # la conversation démarre
```

2. Lui confier une tâche vérifiable

L'agent ne s'arrête que si la commande de vérification réussit.

```
buddy loop "faire passer les tests en échec" --verify-cmd "npm test"
```

3. Le découvrir sans engagement

```
buddy try                # démonstration en 60 secondes
buddy -p "explique le point d'entrée"  # une question, une réponse
buddy cost --latency      # dépense et latence par modèle
```

4. L'application de bureau

Installation séparée, Node.js 22 ou plus récent.

```
buddy install-gui
```

Le paquet publié peut être en retard d'un ou deux jours sur le dépôt : vérifiez `buddy --version` après installation.

Ce qui n'est pas encore prêt

Nous préférons l'écrire noir sur blanc plutôt que de le faire découvrir après la signature.

- Seules les vérifications d'intégration continue sous Linux sont bloquantes ; macOS et Windows tournent au mieux, et l'exécution dans un terminal interactif sous macOS est un problème ouvert connu.
- La boucle vérifiée suppose un modèle qui appelle réellement des outils : un très petit modèle peut caler.
- La voix, la vidéo et les fonctions robot demandent des binaires locaux supplémentaires, qui ne viennent pas avec le paquet npm.
- L'application de bureau Cowork est une installation séparée, qui demande Node.js 22 ou plus récent.
- La flotte, ce sont deux processus et un jeton d'authentification à poser, pas une simple option.

Licence, dépôt et contact

Licence

Business Source License 1.1 : usage interne libre, fourniture du produit en service à des tiers soumise à licence. Bascule automatique en Apache 2.0 le 31 août 2030.

Éditeur

AGILE UP — Structure d'ingénierie logicielle indépendante, Vincennes.

Où le trouver

 github.com/phuetz/code-buddy (<https://github.com/phuetz/code-buddy>)

 `npm i -g @phuetz/code-buddy` (<https://www.npmjs.com/package/@phuetz/code-buddy>)

 [Documentation du projet](https://phuetz.github.io/code-buddy/) (<https://phuetz.github.io/code-buddy/>)

→ Code Buddy dans le studio IA

Contact

contact@agile-up.com
agile-up.com