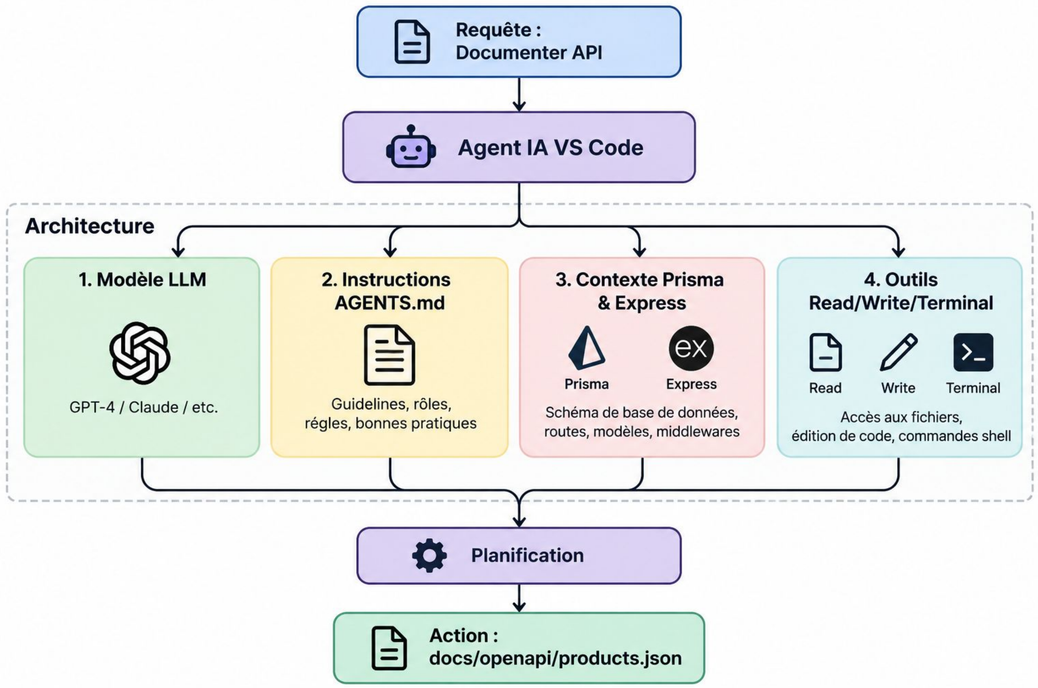
Fiche récap : Créez votre Agent IA

Développement

1. Analyser les fondements et la boucle d'action d'un agent IA dans VS Code.
2. Transformer AGENTS.md en spécification exécutable avec rôle, permissions et critères d'acceptation.
3. Sélectionner un contexte ciblé pour préserver la fenêtre de contexte et limiter les hallucinations.
4. Configurer un serveur MCP local et protéger les secrets avec des variables d'environnement.
5. Valider les productions de l'agent puis les intégrer via une Pull Request et une revue humaine.



Définitions

Agent IA
Système autonome contrôlé capable de lire le projet, d'utiliser des outils, d'agir et de vérifier ses résultats.

Assistant conversationnel
IA passive qui répond à du texte sans agir directement dans l'environnement de développement.

Boucle agentique
Cycle itératif « Comprendre → Agir → Vérifier → Itérer » utilisé par l'agent pour produire et corriger un livrable.

AGENTS.md
Fichier à la racine du dépôt définissant les instructions transversales et la spécification exécutable de l'agent.

Fenêtre de contexte
Quantité limitée d'informations que le modèle peut exploiter simultanément pour raisonner.

MCP
Standard ouvert (Model Context Protocol) permettant à l'agent de se connecter à des outils et à des sources de données externes via des serveurs MCP.

Critères d'acceptation
Règles mesurables permettant à l'agent de déterminer si son livrable respecte les exigences définies.

Bonnes pratiques

- ❏ Restreindre les droits d'écriture de l'agent au dossier /docs/openapi.
- ❏ Fournir explicitement les contrôleurs et le schéma Prisma nécessaires au contexte.
- ❏ Définir des critères d'acceptation mesurables et vérifiables par un linter.
- ❏ Exécuter le linter OpenAPI avant de considérer une tâche comme terminée.
- ❏ Utiliser ${workspaceFolder} pour conserver une configuration MCP portable.
- ❏ Protéger les secrets avec .env, .gitignore, un fichier .env.example, la section inputs de mcp.json et des valeurs anonymisées.
- ❏ Auditer le diff Git pour vérifier Prisma, les codes HTTP et l'absence de secrets.
- ❏ Mettre à jour AGENTS.md avec les évolutions et retours de revue du projet.

Erreurs classiques

- ➢ Valider le travail de l'agent après l'écriture sans exécuter l'étape de vérification.
- ➢ Autoriser l'agent à modifier globalement les dossiers src/ ou prisma/.
- ➢ Omettre schema.prisma du contexte et favoriser des structures de données hallucinées.
- ➢ Injecter node_modules, les fichiers de build ou les logs dans la fenêtre de contexte.
- ➢ Formuler des critères vagues impossibles à contrôler automatiquement.
- ➢ Inscrire un jeton d'API, un secret JWT ou un mot de passe en dur dans le dépôt Git.
- ➢ Committer aveuglément une production générée par l'agent sans relire le diff Git.
- ➢ Pousser directement les productions de l'agent sur main sans passer par une Pull Request.