

GUIDE GRATUIT · SALES EXPERIENZ

Jev + Claude : le modèle qui décide *et le modèle qui écrit*

Pourquoi une IA qui ne sait pas écrire peut trancher à ta place des centaines de fois par jour, et comment la brancher à Claude pas à pas.

Par Laurent Guyonvarch

Sales Experienz — stratégie commerciale augmentée par l'IA

Une IA qui ne sait pas écrire une seule phrase.

Elle ne rédige ni e-mail, ni résumé, ni publication. Et c'est justement pour ça qu'elle vaut le détour.

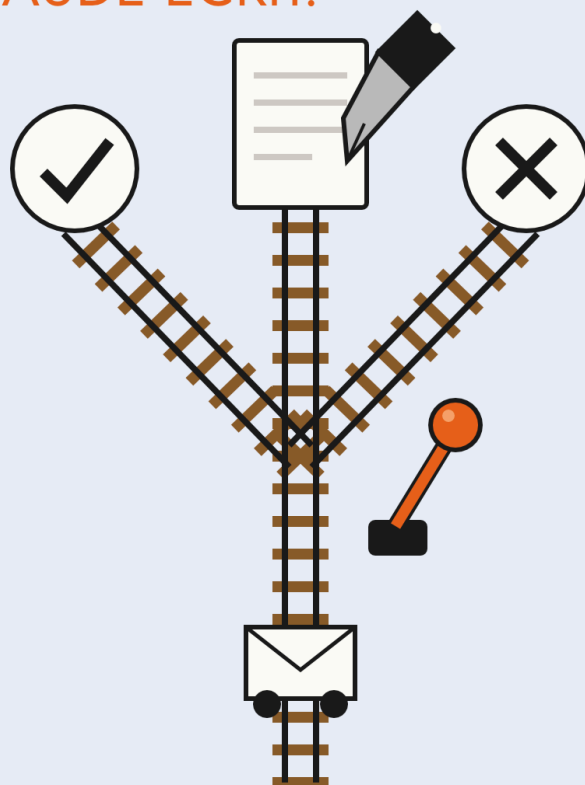
Elle s'appelle Jev. Elle fait une seule chose : décider. Tu lui donnes un message, une fiche client ou une ligne de tableau, et une question. Elle répond en une fraction de seconde, en choisissant dans une liste que tu as écrite, avec un chiffre qui dit à quel point elle est sûre d'elle.

Pense à ta journée. Combien de décisions prends-tu qui ne demandent aucune phrase ? Ce message, c'est un devis ou une réclamation ? Ce prospect, chaud ou froid ? Cette facture, quelle catégorie ? Aujourd'hui, soit tu tranches à la main, soit tu demandes à un grand modèle comme Claude. Il écrit très bien, mais tu paies un rédacteur pour cocher des cases.

Le bon montage utilise les deux. Jev décide, Claude écrit. Ce guide te montre comment les brancher ensemble, pas à pas, même si tu n'as jamais écrit une ligne de code. Puis il te donne 25 consignes prêtes à copier, adaptées à des TPE françaises.

Il a un seul but : qu'à la fin de ta lecture, Jev tourne sur une vraie tâche de ton entreprise.

JEV TRANCHE.
CLAUDE ÉCRIT.



Le modèle qui décide, et celui qui écrit.

Tu ne sais pas encore où l'IA te ferait gagner du temps ? Le Radar IA est un questionnaire en ligne, en accès libre et sans inscription, qui repère où ton entreprise peut gagner du temps avec l'IA. [Ouvrir le Radar IA →](#)

Au programme

- 01** Ce que c'est, ses trois formes de réponse, son prix ramené à une TPE, la confiance et ses limites.

- 02** Six chemins, dont un recommandé en quatre étapes, la commande de test et la sécurité de ta clé.

- 03** Du tri des demandes clients au garde-fou d'assistant, chaque cas avec sa consigne à copier dans Claude Code.

- 04** Les habitudes qui font marcher chaque consigne et les erreurs qui font dérailler un tri.

- 05** Français, prix, code, données, et les adresses de chaque fait cité.

Comprendre Jev en 5 minutes

CE QUE C'EST

Un modèle qui ne fait qu'une chose : trancher

Jev est un modèle d'IA construit pour décider, et pour rien d'autre.

Il est fabriqué par TypeSafe AI, une société de San Francisco fondée en 2024 par Diogo Almeida, Erik Gafni et Sasha Sheng. Diogo Almeida a travaillé chez OpenAI sur les méthodes qui ont rendu les modèles capables de suivre des instructions. Sortie publique : le 15 septembre 2026, avec une levée de 40 millions de dollars.

Le nom vient de l'économiste William Stanley Jevons. TypeSafe appelle son modèle « System One », en référence au système 1 décrit par Daniel Kahneman : la pensée rapide et intuitive, par opposition au raisonnement lent du système 2.

Le fonctionnement tient en deux mots. Tu envoies à Jev un **état**, c'est-à-dire le texte ou les données à juger (un e-mail, une fiche client, une ligne de tableau), et des **questions typées**, c'est-à-dire des questions dont la forme de réponse est fixée d'avance. Il répond à toutes les questions en même temps, en une seule passe, avec des probabilités. Il n'écrit rien d'autre.

Version actuelle : jev-1.13. Le nom exact change selon l'endroit : jev-1.13.0 chez TypeSafe, typesafe/jev-1.13 chez OpenRouter, typesafe-ai/jev sur la passerelle Vercel.

Les trois formes de réponse

Choice : un choix dans ta liste

Jev choisit une option parmi celles que tu fixes, jusqu'à 255. Il renvoie l'option, la probabilité de chaque option et un indice de confiance de 0 à 1. Exemple : un cabinet comptable range chaque e-mail entrant en « pièce comptable », « question fiscale », « prise de rendez-vous » ou « autre ».

Score : une place sur une échelle

Jev place le texte sur une échelle de niveaux que tu décris toi-même, par exemple « calme », « agacé », « très en colère ». Il renvoie la position, les probabilités et la confiance. Exemple : un restaurant classe ses avis clients par niveau de mécontentement pour répondre d'abord aux plus fâchés.

Noul : vrai ou faux

Jev renvoie la probabilité que ce soit « oui », de 0 à 1. Proche de 0,5, il hésite : traite ce cas comme un « je ne sais pas ». Exemple : un organisme de formation juge chaque message reçu avec l'affirmation « La personne demande à s'inscrire à une session. »

Deux détails qui changent tout dans la pratique.

Tu peux poser plusieurs questions dans le même appel. Elles sont traitées en parallèle, sans se voir entre elles, ce qui va plus vite et coûte moins cher que plusieurs appels. Dans les exemples officiels, une question Noul s'écrit comme une affirmation à juger : « Le client exprime une urgence. »

Jev ne sort jamais de ta liste. Il ne peut pas inventer une catégorie que tu n'as pas prévue. Il peut, en revanche, choisir la mauvaise option dans ta liste. Toute la suite du guide sert à repérer ces erreurs avant qu'elles coûtent.

LE PARTAGE DES RÔLES

Jev décide, Claude écrit

La règle tient en quatre mots, et elle vient de TypeSafe lui-même.

Jev trie, note, étiquette et choisit, en une fraction de seconde. Claude rédige les réponses, le code et tout ce qui demande une vraie phrase. Le code, ou ton tableur, fait les calculs et applique les règles fixes. Une personne tranche ce qui reste douteux.

Prends un message client qui arrive. Voici les deux façons de le traiter.

Le même message client, deux montages

ANCIEN MODÈLE

Un message client arrive



Un grand modèle lit tout le message



Il trie, juge, calcule et rédige d'un bloc



**Tu paies un rédacteur pour
cocher des cases, sans savoir à
quel point il est sûr**

NOUVEAU MODÈLE

Le code filtre et fait les calculs



Jev décide : catégorie, urgence, indice
de confiance



Claude écrit la réponse quand il en
faut une



**Un humain voit seulement les cas
douteux**

Jev ou Claude ? Qui fait quoi

La tâche	Qui s'en charge	Pourquoi
Dire si un e-mail est un devis, une réclamation ou autre chose	Jev (Choice)	Un choix dans ta liste, en une fraction de seconde
Noter l'urgence d'une demande de dépannage	Jev (Score)	Une position sur une échelle que tu décris
Dire si un client parle de résilier	Jev (Noul)	Vrai ou faux, avec une probabilité
Rédiger la réponse au client	Claude	Il faut de vraies phrases, et Jev n'écrit pas
Proposer les catégories de départ	Claude	Jev n'invente pas de catégories, il choisit dans les tiennes
Calculer un total, compter des lignes, comparer deux dates	Le code ou le tableur	Jev ne calcule pas et ne compare pas les dates
Écrire le programme qui appelle Jev	Claude Code	Jev ne remplace pas le modèle de Claude Code
Trancher un cas où Jev hésite, ou une décision qui engage	Toi ou une personne de l'équipe	Confiance basse ou risque élevé : un humain décide

Jev trie. Claude rédige. Le code compte. Et toi, tu tranches les cas douteux.

LE PRIX

Ramené à ton quotidien, presque rien

60 000

décisions de ce type pour environ 1 euro (test réel du
11 octobre 2026)

Jev se paie à la lecture, et ses réponses sont gratuites.

Prix affiché : 0,042 dollar par million de **jetons** lus, soit environ 4 centimes d'euro. Un jeton, c'est à peu près les trois quarts d'un mot.

Le 11 octobre 2026, j'ai envoyé à Jev un message client de deux lignes avec deux questions. Total : 397 jetons, soit environ 0,0015 centime d'euro. Avec 1 euro, tu paies environ 60 000 décisions de ce type.

Imagine un chauffagiste qui reçoit 30 demandes par jour ouvré, environ 660 par mois. Les faire trier par Jev lui coûterait de l'ordre d'un centime d'euro par mois. Les réponses rédigées par Claude, elles, se paient à part.

Côté vitesse, TypeSafe annonce de 70 à 500 millisecondes par appel, et OpenRouter mesure 0,18 seconde en médiane. TypeSafe affiche aussi « 193,6 fois plus rapide, 444,6 fois moins cher » que les grands modèles : ce chiffre vient de ses propres tests, sans vérification indépendante.

LA CONFIANCE

Le chiffre qui dit qui doit agir

Chaque réponse arrive avec un indice de confiance. C'est lui qui aiguille.

Confiance haute : l'automatisation agit seule. Confiance moyenne : on fait vérifier.
Confiance basse : un humain décide.

Le seuil dépend du risque. Ranger un e-mail dans le mauvais dossier se rattrape en deux clics. Envoyer un devis ou relancer un client à tort se rattrape beaucoup moins bien. Plus l'action est grave, plus la barre monte.

Et garde toujours un plancher à 0,5 : il attrape ce que Jev dit lui-même ne pas savoir.

LES LIMITES

Ce que Jev fait bien, ce qu'il fait mal

D'après la documentation officielle de TypeSafe

Il le fait bien

- Choisir une option dans une liste que tu fixes, jusqu'à 255 options
- Placer un texte sur une échelle que tu décris
- Répondre vrai ou faux avec une probabilité
- Répondre à plusieurs questions dans le même appel
- Dire à quel point il est sûr de lui

Il le fait mal

- Une question floue : il lit au pied de la lettre
- Calculer, compter, comparer des dates
- Les doubles négations et les raisonnements en plusieurs étapes
- Un texte trop long, rempli de détails inutiles
- Un texte écrit pour le manipuler
- Des instructions et des options qui se contredisent
- L'ordre des options : il penche un peu vers la première
- Écrire : pour rédiger, c'est Claude

Deux limites de taille et de format.

Un appel accepte 64 000 jetons au maximum, dont 32 000 pour l'état et la question la plus longue. Largement de quoi juger un e-mail ou une fiche client, pas un classeur entier.

Jev lit du texte seulement. Une photo de chantier, un message vocal ou une vidéo doivent d'abord être décrits en texte avant de lui être envoyés.

LE PIÈGE DU FRANÇAIS

L'anglais est la langue où Jev est le plus précis. Le français marche, « mais pas aussi bien », selon les mots de TypeSafe. Avant de lui confier un tri, teste-le sur tes propres exemples en français. Tu verras en partie 4 comment faire avec 50 exemples.

À FAIRE MAINTENANT

Choisis une décision que tu prends plusieurs fois par jour sans écrire une phrase : trier un message, qualifier un prospect, classer une dépense. Écris-la sur une feuille sous forme de question, avec ta liste d'options, une phrase de description pour chacune, et une option « Autre ». Tu t'en serviras dès la partie 2.

PARTIE 2

Installation : brancher Jev à Claude

Une précision avant de taper la moindre commande : Jev ne devient pas le cerveau de Claude Code.

TypeSafe le dit noir sur blanc : Jev ne remplace pas le modèle qui fait tourner Claude Code. Aucun réglage ne transforme Claude Code en « Claude Code propulsé par Jev ».

Le partage est simple. Claude Code est le chef de chantier : il écrit le programme. Ce programme, une fois en place, appelle Jev autant de fois qu'il le faut pour trancher, sans que tu aies à rouvrir Claude Code.

Brancher Jev à Claude revient donc à faire deux choses : donner à ton ordinateur une clé qui ouvre l'accès à Jev, et apprendre à Claude Code comment l'appeler. Le chemin 1 fait les deux en quatre étapes.

Cinq mots à connaître avant de commencer

Terminal

L'application de ton Mac où l'on tape des commandes au lieu de cliquer. Tu y colles une ligne, tu appuies sur Entrée, l'ordinateur l'exécute.

Clé d'accès

Une longue suite de caractères que te donne OpenRouter ou TypeSafe. Elle prouve que c'est toi et sert à te facturer : traite-la comme le code de ta carte bancaire.

Variable d'environnement

Une étiquette que ton ordinateur garde en mémoire, ici `OPENROUTER_API_KEY`, avec ta clé dedans. Les programmes la lisent sans que la clé soit écrite dans leur code.

Skill

Un mode d'emploi que Claude Code charge pour une tâche précise. Le skill officiel de TypeSafe lui apprend à travailler avec Jev.

Nœud n8n

Une brique dans n8n, l'outil d'automatisation. Chaque nœud fait une action (lire un e-mail, appeler Jev, envoyer un message) et on relie les nœuds entre eux.

Les six chemins pour accéder à Jev

Chemin	Pour qui	Ce qu'il faut	Difficulté
0. Le Playground de TypeSafe	Tu veux voir Jev répondre avant tout	Un navigateur, se connecter à la console TypeSafe	Très facile
1. Claude Code + skill officiel + clé OpenRouter (recommandé)	Le dirigeant qui veut construire ses propres outils	Claude Code, le Terminal, une clé OpenRouter avec un peu de crédit	Facile
2. Appel direct par OpenRouter	Vérifier que tout marche, ou appeler Jev depuis un nœud HTTP de n8n	Le Terminal et ta clé OpenRouter	Facile si tu copies la commande
3. Compte TypeSafe en direct	Un développeur, ou toi si tu veux un compte chez TypeSafe	Une clé TypeSafe, Python 3.10 ou plus, ou Node 20 ou plus	Moyenne
4. Passerelle Vercel AI Gateway	Une seule facture pour Claude, Jev et les autres modèles	Clé AI_GATEWAY_API_KEY, Node.js 22.18 ou plus, bibliothèque ai 7.0.128 ou plus	Moyenne
5. n8n	Tes automatisations tournent déjà dans n8n	n8n et le nœud TypeSafe AI, ou un nœud HTTP Request	Facile à moyenne

Si tu hésites : chemin 0 pour voir, chemin 1 pour construire.

Chemin 0 : essayer sans rien installer

Avant d'installer quoi que ce soit, regarde Jev répondre.

Va sur console.typesafe.ai/playground et connecte-toi. Colle un vrai message client comme état, ajoute la question que tu as écrite à la fin de la partie 1, et regarde la réponse arriver.

Fais trois essais : un cas évident, un cas limite, puis le même cas avec les options dans l'ordre inverse. Si la réponse change avec l'ordre, ta question ou tes descriptions d'options sont à reprendre.

Chemin 1 (recommandé) : Claude Code, le skill officiel et une clé OpenRouter

C'est le chemin que je conseille à un dirigeant qui a déjà Claude Code.

OpenRouter est un service qui donne accès à des centaines de modèles avec un seul compte et une seule clé. Jev y porte le nom `typesafe/jev-1.13`. Tu n'as besoin ni d'un compte TypeSafe, ni de passer par une liste d'attente.

Les commandes ci-dessous sont écrites pour le Terminal d'un Mac. Tu les retrouves dans le fichier à télécharger, prêtes à copier.

Brancher Jev en quatre étapes

- 1 **Étape 1 : créer ta clé OpenRouter**
- 2 **Étape 2 : donner la clé à ton ordinateur**
- 3 **Étape 3 : installer le skill officiel de TypeSafe**
- 4 **Étape 4 : ta première demande**

Les commandes du chemin 1

Recopie chaque commande telle quelle. Seul sk-or-... est à remplacer par ta clé.

PROMPT 01 Étape 2 : donner la clé à ton ordinateur (Terminal)

```
export OPENROUTER_API_KEY=sk-or-...
```

PROMPT 02 Étape 3 : installer le skill dans Claude Code (Terminal, 2 commandes)

```
claude plugin marketplace add typesafe-ai/skills  
claude plugin install typesafe@typesafe-ai
```

PROMPT 03 Étape 3, variante : dans un autre agent (Codex, Cursor...)

```
npx skills add typesafe-ai/skills --skill typesafe-ai
```

PROMPT 04 Étape 3 : appeler le skill dans Claude Code

```
/typesafe:typesafe-ai
```

PROMPT 05 Étape 4 : la consigne de départ recommandée par TypeSafe (traduite)

Utilise le skill TypeSafe. Regarde mon projet et dis-moi où un appel à Jev rendrait les choses plus rapides ou moins chères. Pose-moi des questions avant de construire.

TypeSafe donne quatre conseils pour travailler avec un agent sur Jev. Je les applique tels quels.

Relis le plan avant de laisser construire. Range les questions et les seuils dans un seul fichier, pour pouvoir les relire d'un coup d'œil.

Les agents écrivent mal les questions : retravaille-les à deux avec Claude Code, en lui demandant de décrire chaque option en une phrase. Et ne crois aucune affirmation sans la faire vérifier, même quand Claude Code annonce que tout marche.

Vérifier que ça marche

Une commande suffit pour savoir si ta clé et Jev se parlent.

Dans la fenêtre du Terminal où tu as tapé export, colle la commande ci-dessous. Elle appelle Jev directement par OpenRouter, avec un vrai message client et deux questions : quel service doit le traiter, et le client exprime-t-il une urgence ?

Le morceau <<'EOF' sert à écrire des apostrophes dans le message sans casser la commande. Ne le touche pas.

La commande de test

À coller telle quelle dans le Terminal. Ta clé est lue dans la variable OPENROUTER_API_KEY, tu n'as rien à remplacer.

PROMPT 01 Test de branchement par OpenRouter (chemin 2)

```
curl https://openrouter.ai/api/alpha/decisions \
-H "Authorization: Bearer $OPENROUTER_API_KEY" \
-H "Content-Type: application/json" \
-d @- <<'EOF'
{
  "model": "typesafe/jev-1.13",
  "state": "Bonjour, je n'arrive pas à payer ma facture depuis trois jours, le lien de paiement ne marche pas. C'est urgent.",
  "questions": {
    "service": {
      "type": "choice",
      "instructions": "Quel service doit traiter ce message ?",
      "criteria": {
        "facturation": "Paiement, facture, remboursement",
        "technique": "Bug, site, connexion",
        "commercial": "Prix, devis, nouvelle commande",
        "autre": "Rien de tout cela"
      }
    },
    "urgent": {
      "type": "noul",
      "instructions": "Le client exprime une urgence."
    }
  }
}
EOF
```

Ce que tu dois voir revenir.

J'ai posé exactement ce message et ces deux questions à Jev le 11 octobre 2026, par la passerelle Vercel du chemin 4. Réponse : « facturation » à 87 % (technique à 13 %), urgence à 98 %, pour 397 jetons.

Si ta réponse contient un service choisi et une probabilité pour l'urgence, c'est branché. Chaque réponse d'OpenRouter contient aussi `usage.cost` : le prix de l'appel, en dollars. Regarde-le une fois, ça remet les idées en place sur le coût.

Cette même adresse, POST <https://openrouter.ai/api/alpha/decisions>, sert aussi à appeler Jev depuis un nœud HTTP Request de n8n. C'est le chemin 2.

Chemin 3 : un compte TypeSafe en direct

Pour un développeur, ou si tu préfères un compte chez TypeSafe.

Crée ta clé sur console.typesafe.ai/keys et range-la dans la variable `TYPESAFE_API_KEY`. L'adresse de Jev est alors `POST https://api.typesafe.ai/v1/systemone`. Deux kits existent : Python (version 3.10 ou plus) et JavaScript (Node 20 ou plus).

L'exemple ci-dessous pose les deux mêmes questions que le test. Le kit sait aussi passer par OpenRouter avec ta clé OpenRouter : c'est le dernier élément du fichier.

Les commandes du chemin 3

À recopier à l'identique.

PROMPT 01 Installer le kit Python

```
pip install typesafe-sdk
```

PROMPT 02 Installer le kit JavaScript

```
npm install @typesafe-ai/sdk
```

PROMPT 03 Exemple en Python

```
from typesafe_sdk import Choice, Noul, TypeSafeClient

client = TypeSafeClient()
reponse = client.system_one(
    state="Bonjour, je n'arrive pas à payer ma facture depuis trois jours. C'est urgent.",
    questions={
        "service": Choice(
            instructions="Quel service doit traiter ce message ?",
            criteria={
                "facturation": "Paieement, facture, remboursement",
                "technique": "Bug, site, connexion",
                "commercial": "Prix, devis, nouvelle commande",
                "autre": "Rien de tout cela",
            },
        ),
        "urgent": Noul(instructions="Le client exprime une urgence."),
    },
)
print(reponse.answers["service"].choice, reponse.answers["service"].confidence)
print(reponse.answers["urgent"].noul)
```

PROMPT 04 Variante : le kit TypeSafe qui passe par OpenRouter

```
import os
from typesafe_sdk import TypeSafeClient

client = TypeSafeClient(
    api_key=os.environ["OPENROUTER_API_KEY"],
    base_url="https://openrouter.ai/api",
)
```

Chemin 4 : la passerelle Vercel AI Gateway

C'est celle que j'utilise chez Sales Experiencz : une seule facture pour Claude, Jev et les autres modèles.

Il te faut la bibliothèque ai en version 7.0.128 ou plus, Node.js 22.18 ou plus, et ta clé dans la variable AI_GATEWAY_API_KEY. Sur cette passerelle, Jev s'appelle typesafe-ai/jev.

Trois temps : installer la bibliothèque, créer le fichier decision.mts, le lancer.

Les commandes du chemin 4

À recopier à l'identique, sauf ta clé. Le troisième élément est le contenu du fichier decision.mts.

PROMPT 01 Donner ta clé de la passerelle à ton ordinateur (Terminal)

```
export AI_GATEWAY_API_KEY=ta_cle_de_passerelle
```

PROMPT 02 Installer la bibliothèque

```
npm install ai@latest
```

PROMPT 03 Contenu du fichier decision.mts

```
import { experimental_decide as decide } from 'ai';

const resultat = await decide({
  model: 'typesafe-ai/jev',
  state: "Bonjour, je n'arrive pas à payer ma facture depuis trois jours. C'est urgent.",
  questions: {
    urgent: { type: 'boolean', instructions: 'Le client exprime une urgence.' },
  },
});

console.log(JSON.stringify(resultat.answers, null, 2));
```

PROMPT 04 Lancer le test

```
node decision.mts
```

LE PIÈGE QUE J'AI VÉCU

Sur la passerelle Vercel, le type oui/non s'appelle boolean, et pas noul, et la réponse donne probability. Pour une question Choice, la passerelle ne renvoie pas toujours de confidence : prends alors la probabilité de l'option choisie.

Chemin 5 : n8n

Si tes automatisations tournent déjà dans n8n, Jev s'y branche sans code.

Il existe un nœud officiel « TypeSafe AI » dans n8n, construit par TypeSafe et vérifié par n8n. Il a deux actions. Evaluate pose tes questions et récupère les réponses. Route envoie l'élément vers la sortie qui correspond au choix de Jev, comme un nœud Switch.

Le montage typique : une sortie « confiance haute » qui continue toute seule, une sortie « confiance basse » qui part vers Claude ou vers une personne. Sinon, un nœud HTTP Request vers l'adresse OpenRouter du chemin 2 fait la même chose.

LA SÉCURITÉ DE TA CLÉ

Ta clé d'accès reste sur ton ordinateur ou sur ton serveur. Elle ne va jamais dans le code d'une page web, ni dans un fichier partagé, ni dans un message, ni dans une conversation avec une IA. Mets peu de crédit au départ : quelques euros suffisent pour des dizaines de milliers de décisions, et c'est aussi ce que tu risques au pire si la clé fuit. Enfin, range tes questions et tes seuils dans un seul fichier, relu par un humain avant chaque mise en service.

C'est branché quand...

- ☐ Ta clé OpenRouter est créée et ton compte a quelques euros de crédit
- ☐ La commande export a été tapée dans le Terminal, et Claude Code est lancé depuis cette même fenêtre
- ☐ Claude Code réagit quand tu écris /typesafe:typesafe-ai ou « utilise le skill TypeSafe »
- ☐ La commande de test renvoie un service choisi et une probabilité pour l'urgence
- ☐ Tu as regardé usage.cost au moins une fois
- ☐ Ta clé n'apparaît dans aucun fichier partagé, aucune page web, aucune conversation
- ☐ Tes questions et tes seuils sont rangés dans un seul fichier que tu sais relire

À FAIRE MAINTENANT

Suis le chemin 1 jusqu'au bout, puis lance la commande de test. Ensuite, remplace le message de test par un vrai message reçu cette semaine, et les quatre services par la liste que tu as écrite à la fin de la partie 1. Si la réponse tient debout, tu es prêt pour la partie 3.

PARTIE 3

25 cas pratiques, une consigne chacun

Chaque cas tient en trois phrases et une consigne à coller dans Claude Code.

Les cas sont rangés en quatre groupes : facile, intermédiaire, avancé, puis des cas trouvés en ligne au-delà de notre source principale. Les idées des 19 premiers viennent d'un guide anglais publié par RoboNuggets. Je les ai adaptées à des entreprises françaises.

Dans chaque consigne, remplace ce qui est entre crochets [] par tes propres éléments. La dernière ligne sert de filet : si Claude Code n'a pas encore Jev sous la main, elle lui dit comment le trouver.

Chaque groupe de consignes est réuni dans un fichier à télécharger. Commence par un seul cas, celui qui te fait gagner le plus de temps cette semaine.

NIVEAU FACILE

Tu pointes Jev sur ce que tu as déjà, il trie

Cas 1 à 8

1. Jev dans Google Sheets

Un cabinet comptable reçoit chaque mois les relevés de ses clients. Tu tapes « Catégorie » en haut d'une colonne, et Jev remplit chaque ligne en choisissant dans ta liste. Il ne sort jamais de cette liste, donc pas de catégorie surprise.

2. Trier les demandes clients

Un chauffagiste reçoit des demandes de devis, de dépannage, des questions de facture. Jev range chaque message et dit à quel point il est sûr. Les cas faciles partent dans la bonne automatisation, les autres arrivent chez une personne.

3. Lire les publicités de tes concurrents

Une agence immobilière veut savoir ce que publient ses concurrents sur Facebook et Instagram. Une extension Chrome enregistre leurs annonces dans la Bibliothèque publicitaire de Meta, et Jev étiquette le format, l'appel à l'action et l'étape visée. Tu obtiens un tableau de ce qui tourne dans ton secteur.

4. Tirer des extraits d'une longue vidéo

Un organisme de formation enregistre un webinaire d'une heure. Jev note chaque passage possible de 30 à 60 secondes et te rend les dix meilleurs, avec le début et la fin. Tu enregistres une fois, Jev trouve les moments à publier.

5. Repérer les clients sur le départ

Un institut de beauté vend des abonnements annuels. Jev lit ton export clients et place chacun dans « en bonne santé », « à surveiller » ou « risque de départ ». Le code calcule l'ancienneté avant, puisque Jev ne compare pas les dates.

6. Relier les pages de ton site entre elles

Un menuisier publie des articles sur son site. Jev lit chaque page et choisit vers quelles autres pages elle doit renvoyer, et sur quelle expression. Les visiteurs trouvent leur chemin, et ton référencement en profite un peu.

7. Trouver les acheteurs dans tes commentaires

Une boutique de déco vend beaucoup par Instagram. Un outil comme Apify récupère les commentaires, et Jev étiquette chacun : prêt à acheter, question, réclamation ou autre. Les acheteurs remontent en haut de la liste au lieu de rester noyés.

8. Vérifier Jev avant de lui faire confiance

Tu as déjà trié à la main des centaines de messages, de dépenses ou de fiches. Jev refait le tri sans voir tes réponses, et tu compares. Les désaccords te disent quelles options ou quelles consignes reprendre.

Les consignes du niveau facile

À coller dans Claude Code, une à la fois. Remplace ce qui est entre crochets.

PROMPT 01 1. Jev dans Google Sheets

Construis-moi un module complémentaire Google Sheets. Quand je tape un nouvel en-tête de colonne, par exemple « Catégorie », envoie chaque ligne avec cet en-tête à Jev et remplis la colonne avec ses réponses. Laisse-moi donner une liste d'options autorisées, par exemple [Fournitures, Logiciels, Déplacements, Restaurant, Autre]. Ajoute une colonne avec la confiance de Jev, et colore en orange les lignes sous [0,6].

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 02 2. Trier les demandes clients

Construis-moi une automatisation pour les demandes de mes clients, qui arrivent par [mon adresse e-mail / le formulaire de mon site]. Pour chaque nouveau message, demande à Jev de choisir la catégorie parmi [devis, dépannage, facture, autre], avec une phrase de description pour chaque option, et récupère sa confiance. Envoie les messages traitables automatiquement vers [l'action prévue], et tout ce qui est sous [60] % de confiance à [la personne qui s'en occupe]. Range ce seuil dans un seul fichier pour que je puisse le changer facilement.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 03 3. Lire les publicités de tes concurrents

Construis-moi une extension Chrome pour la Bibliothèque publicitaire de Meta. Quand je parcours les annonces d'un concurrent, enregistre chaque annonce dans une base de données dans l'extension. Utilise Jev pour étiqueter son format [image, vidéo, carrousel], son appel à l'action et l'étape visée (découverte, réflexion ou achat). Ajoute un bouton pour tout exporter en CSV.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 04 4. Tirer des extraits d'une longue vidéo

Transcris ma vidéo [webinaire.mp4] avec le minutage de chaque mot. Découpe-la en extraits possibles de 30 à 60 secondes. Utilise Jev pour noter chaque extrait sur une échelle de [faible / correct / très bon pour une vidéo courte], en décrivant chaque niveau. Donne-moi les dix meilleurs, avec l'heure de début et de fin. La transcription est en français : signale-moi les extraits où Jev hésite.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 05 5. Repérer les clients sur le départ

Voici mon export clients, [clients.csv], avec la date d'inscription, la formule, la date de dernière visite et une colonne qui dit si la personne a commencé à résilier. Calcule d'abord dans le code l'ancienneté et le nombre de jours depuis la dernière visite, puisque Jev ne compare pas les dates. Ensuite, utilise Jev pour placer chaque client dans « en bonne santé », « à surveiller » ou « risque de départ », avec une description de chaque niveau. Trie les clients à risque en haut et garde la confiance de Jev dans une colonne.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 06 6. Relier les pages de ton site entre elles

Lis toutes les pages de mon site à partir de [l'adresse de mon plan du site]. Pour chaque page, utilise Jev pour choisir vers quelles autres pages elle doit renvoyer, et sur quelle expression. Donne-moi la liste des liens à ajouter, les plus utiles en premier. Ne modifie rien sur le site : je relis la liste avant.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 07 7. Trouver les acheteurs dans tes commentaires

Utilise Apify pour récupérer les [500] derniers commentaires du compte [Instagram ou Facebook de ma boutique]. Utilise Jev pour étiqueter chaque commentaire : prêt à acheter, question, réclamation ou autre, avec une phrase de description pour chaque option. Donne-moi une liste avec les commentaires « prêt à acheter » en haut, et le lien vers chacun.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 08 8. Vérifier Jev avant de lui faire confiance

J'ai [exemples-tries.csv] avec des exemples qu'une personne a déjà triés à la main. Coupe-les en deux : la première moitié sert à retoucher mes catégories et mes consignes, et tu donnes les résultats seulement sur la seconde moitié. Fais passer chaque exemple dans Jev sans lui montrer la bonne réponse, et ajoute sa réponse et sa confiance à côté du tri humain. Montre-moi combien de fois ils sont d'accord, où ils divergent le plus, et la précision de Jev par tranche de confiance. Refais le test avec les options dans l'ordre inverse.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

NIVEAU INTERMÉDIAIRE

Jev s'installe dans tes outils et tourne à chaque fois

Cas 9 à 14

9. Laisser Jev choisir le bon skill

Tu as accumulé des dizaines de skills dans Claude Code. Ta demande arrive, la liste de tes skills sert de menu, et Jev choisit celui que Claude charge. Une option « Aucun » évite un mauvais choix forcé.

10. Laisser Jev choisir le bon modèle

Chercher un fichier ne demande pas le modèle le plus cher. Jev estime la difficulté de chaque demande et l'envoie au modèle Claude le moins cher capable de la traiter : Haiku, Sonnet ou Opus. Ta facture suit la difficulté réelle du travail.

11. Jev devant ton assistant e-mail

Un consultant indépendant reçoit des dizaines d'e-mails par jour, dont une bonne part sans intérêt. Jev lit chaque e-mail d'abord : faut-il répondre, est-ce une proposition de partenariat, est-ce une arnaque ? Seuls ceux qui demandent une réponse vont chez Claude pour un brouillon.

12. Une extension qui nettoie ton fil LinkedIn

Tu veux lire ton fil sans les publications creuses écrites à la chaîne. À chaque publication qui s'affiche, Jev répond vrai ou faux à une question que tu fixes, et l'extension replie celles qui la remplissent. Un petit bouton les rouvre si tu veux.

13. Une recherche qui comprend ce que tu veux dire

Un client cherche « délai pour annuler » dans tes conditions générales, alors que le texte parle de « droit de rétractation ». Jev découpe la page en paragraphes, choisit celui qui correspond et y amène le lecteur. Les mots n'ont pas besoin de coïncider.

14. Retrouver tes photos par leur contenu

Un paysagiste a des milliers de photos de chantiers aux noms de fichiers illisibles. Un petit modèle bon marché écrit une ligne de description par photo, puisque Jev ne lit que du texte. Tu tapes « terrasse en bois », et Jev choisit les photos qui correspondent.

Les consignes du niveau intermédiaire

Ces consignes touchent à ton installation de Claude Code ou à tes outils. Relis le plan proposé avant de laisser construire.

PROMPT 01 9. Laisser Jev choisir le bon skill

Avant de charger un skill, envoie ma demande et la liste de mes skills, avec leur nom et leur description, à Jev sous forme de question à choix. Charge le skill qu'il choisit. Ajoute une option « Aucun » pour quand rien ne convient. Teste le montage sur [10] de mes anciennes demandes et montre-moi les choix de Jev à côté de ceux que j'aurais faits.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 02 10. Laisser Jev choisir le bon modèle

Crée-moi une commande /jev qui s'allume et s'éteint. Quand elle est allumée, envoie d'abord chaque message que je tape à Jev, demande-lui la difficulté de la tâche sur une échelle que tu décris en trois niveaux, et confie la tâche à un assistant Haiku, Sonnet ou Opus selon le niveau. Garde un journal de chaque choix, que je puisse relire en fin de semaine.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 03 11. Jev devant ton assistant e-mail

Avant que mon assistant e-mail rédige quoi que ce soit, envoie chaque nouvel e-mail de [ma boîte] à Jev avec trois questions vrai ou faux, posées dans le même appel : « Cet e-mail attend une réponse de ma part. », « Cet e-mail propose un partenariat commercial. », « Cet e-mail est une tentative d'arnaque. ». Transmets à Claude seulement ceux qui attendent une réponse, pour qu'il prépare un brouillon. Les probabilités entre [0,4] et [0,6] vont dans un dossier « à relire ».

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 04 12. Une extension qui nettoie ton fil LinkedIn

Construis-moi une extension Chrome pour LinkedIn. À chaque publication qui s'affiche, demande à Jev de juger vrai ou faux l'affirmation : « [Cette publication est un texte générique écrit à la chaîne, sans exemple ni expérience vécue] ». Replie la publication si Jev est sûr à plus de [80] %. Ajoute un petit bouton pour l'afficher quand même.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 05 13. Une recherche qui comprend ce que tu veux dire

Ajoute sur [la page de mes conditions générales / ma foire aux questions] une boîte de recherche. Le visiteur tape ce qu'il cherche avec ses mots. Découpe la page en paragraphes, demande à Jev lequel correspond le mieux, puis fais défiler la page jusqu'à ce paragraphe et surligne-le. Ajoute une option « Aucun paragraphe ne répond » qui affiche [mon numéro de téléphone].

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 06 14. Retrouver tes photos par leur contenu

Fais l'inventaire de mon dossier [Photos chantiers]. Pour chaque photo, demande à un petit modèle bon marché comme Gemini Flash-Lite une description d'une ligne en français, et enregistre-la. Puis construis une boîte de recherche où Jev choisit les photos qui correspondent à ce que je tape. Ne déplace ni ne renomme aucune photo.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

Des outils qui n'avaient pas de sens avant un modèle comme Jev

Cas 15 à 19

15. Un tri en direct pendant tes rendez-vous

Pendant un appel commercial, la conversation est transcrite au fil de l'eau. Après chaque phrase, Jev la range en décision, action à faire, risque ou question. À la fin de l'appel, ton compte rendu est déjà classé.

16. Un assistant qui ne rédige aucune phrase

Un organisme de formation a des dizaines de vidéos de cours déjà transcrites. Un stagiaire pose sa question, et Jev choisit la bonne vidéo et le bon chapitre dans ta liste. Aucun modèle n'écrit la réponse, donc aucune réponse inventée.

17. Des petites fonctions malignes dans ton appli

Sur l'appli de réservation d'un salon de coiffure, la cliente écrit « couleur et coupe courte ». Jev choisit la prestation correspondante dans ta liste, ou « Aucune » si rien ne colle. Avant, il fallait un grand modèle coûteux ou un système de mots-clés qui ratait souvent.

18. Une page qui se compose pour chaque visiteur

Une école de langues a des blocs prêts : accroche, tarifs, inscription, témoignages. Selon ce que tu sais du visiteur, Jev choisit quatre blocs et leur ordre. La page s'affiche vite, puisque Jev choisit les blocs sans les écrire.

19. Trouver tes propres cas

Le site shipwithjev.com recense des centaines de réalisations publiques, classées par catégorie. Tu le donnes à Claude avec un fichier qui décrit ton entreprise, et il te dit lesquelles te servent. Puis il relit tes dernières sessions pour repérer les étapes où un appel à Jev aurait suffi.

Les consignes du niveau avancé

Ces projets demandent plusieurs séances avec Claude Code. Fais-les tourner sur des données de test avant de les brancher sur des clients.

PROMPT 01 15. Un tri en direct pendant tes rendez-vous

Construis une page qui écoute ma réunion par le micro et la transcrit en direct, en français. Après chaque phrase, demande à Jev si c'est une décision, une action à faire, un risque, une question ou « autre », et affiche-la sous le bon titre. À la fin, propose un export de la liste. Préviens les participants que la réunion est transcrite.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 02 16. Un assistant qui ne rédige aucune phrase

Voici les transcriptions de mes vidéos de formation, dans [dossier], avec la liste des chapitres et leur minutage. Construis une boîte de discussion où un stagiaire pose une question, et où Jev choisit la meilleure vidéo et le bon chapitre dans ma liste. Affiche la vidéo à la bonne minute, sans aucun texte rédigé par un modèle. Ajoute une option « Aucune vidéo ne répond » qui renvoie vers [mon adresse e-mail].

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 03 17. Des petites fonctions malignes dans ton appli

Dans [mon appli ou mon formulaire de réservation], quand une cliente décrit ce qu'elle veut avec ses mots, envoie sa phrase à Jev avec ma liste de [prestations], chacune avec une phrase de description. Présélectionne la prestation que Jev choisit, et laisse la cliente la changer. Ajoute une option « Aucune » pour quand rien ne correspond.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 04 18. Une page qui se compose pour chaque visiteur

Construis-moi une page d'accueil qui se compose seule. J'ai des blocs pour [l'accroche, les tarifs, l'inscription, les témoignages]. Quand quelqu'un arrive, envoie à Jev ce qu'on sait de lui [page d'origine, recherche tapée], laisse-le choisir quatre blocs et affiche-les dans cet ordre. Garde une version par défaut si Jev ne répond pas.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 05 19. Trouver tes propres cas

Voici le site shipwithjev.com et un fichier qui décrit mon entreprise : [fichier-entreprise.md]. D'après tout ce que tu sais de moi, quelles sont les cinq réalisations avec Jev qui me serviraient le plus ? Ensuite, relis mes [20] dernières sessions et liste les étapes où un appel rapide à Jev aurait fait le travail. Pour chacune, dis-moi si c'est un choix, une note ou un vrai/faux.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

TROUVÉS EN LIGNE

Six cas publiés par TypeSafe, OpenRouter et d'autres

Ces cas viennent de la documentation de TypeSafe, du livre de recettes d'OpenRouter et d'articles parus en septembre et octobre 2026.

La documentation de TypeSafe cite aussi le tri de CV selon des critères écrits. En Europe, l'IA utilisée pour recruter est classée « à haut risque » par l'AI Act. Si tu t'en sers, Jev prépare le tri et un humain décide : il n'écarte jamais seul un candidat.

Cas 20 à 25

20. Valider les demandes de permission de Claude Code

Claude Code te demande sans cesse « je peux lancer cette commande ? ». Jev répond à deux questions vrai ou faux, « la commande est-elle réversible ? » et « sert-elle la tâche en cours ? », et ne valide qu'au-dessus de 90 %. Les commandes dangereuses restent bloquées par une liste fixe avant Jev. Résultats publiés par OpenRouter : lancer les tests (0,93 et 0,95) est validé, ajouter une bibliothèque sans rapport (0,45 et 0,09) reste en attente, une mise en ligne (0,04 et 0,07) est bloquée. Coût mesuré : 0,0000168 dollar par vérification, une infime fraction de centime d'euro.

21. Un garde-fou devant ton assistant de site

Une agence de voyages a un assistant de discussion sur son site. Chaque message entrant et sortant passe par des questions vrai ou faux (tentative de détournement, demande illégale) et une note de gravité. Le code laisse passer, fait relire ou bloque.

22. Vérifier les citations d'un article

Un cabinet d'avocats fait rédiger ses articles de blog par Claude, avec des citations de textes. Jev vérifie que chaque citation dit bien ce qu'on lui fait dire. Dans la démo officielle, sur 8 citations, les 4 justes ont été validées à 0,93 et plus, et les 4 erreurs plantées ont toutes été repérées : un petit test, à refaire sur tes textes.

23. Router tes fiches et tes e-mails dans n8n

Une entreprise de nettoyage reçoit des demandes formulées de mille façons. Dans n8n, Jev choisit le bon service pour chaque demande, trie les e-mails par catégorie et envoie chaque fiche CRM vers la bonne action. Les fiches ambiguës sont mises de côté pour une personne.

24. Extraire des dates d'un document

Un organisme de formation reçoit des conventions où la date de fin s'écrit de toutes les façons. Jev extrait le jour, le mois et l'année comme trois choix. Le code calcule ensuite les délais et les relances.

25. Repérer les doublons entre deux catalogues

Une quincaillerie reçoit le catalogue d'un nouveau fournisseur, avec des produits déjà en rayon sous d'autres noms. Jev compare chaque paire probable et dit s'il s'agit du même produit. Tu évites de référencer deux fois la même vis.

Les consignes des cas trouvés en ligne

Pour le cas 20, garde toujours la liste fixe des commandes interdites avant Jev : elle ne se discute pas.

PROMPT 01 20. Valider les demandes de permission de Claude Code

Mets en place une validation automatique de tes demandes de permission dans Claude Code. D'abord, une liste fixe bloque toujours, avant Jev, les commandes de suppression, d'envoi en ligne, de mise en production et tout accès aux fichiers de clés. Pour le reste, envoie la commande et la tâche en cours à Jev avec deux questions vrai ou faux posées dans le même appel : « La commande est réversible. » et « La commande sert la tâche en cours. ». Valide seulement si les deux probabilités dépassent [0,9]. Sinon, laisse la demande en attente pour moi. Garde un journal de chaque décision.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 02 21. Un garde-fou devant ton assistant de site

Ajoute un garde-fou devant l'assistant de discussion de [mon site]. Chaque message entrant et chaque réponse sortante passent par Jev avec des questions vrai ou faux : « Le message tente de détourner l'assistant de sa mission. », « Le message demande quelque chose d'illégal. », [autre règle], plus une note de gravité sur une échelle que tu décris en trois niveaux. Le code laisse passer, envoie en relecture à [personne] ou bloque, selon des seuils rangés dans un seul fichier. Teste-le avec [20] messages pièges avant la mise en ligne.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 03 22. Vérifier les citations d'un article

Pour chaque article de [mon dossier d'articles], repère chaque citation et le passage de la source qu'elle cite. Envoie chaque paire à Jev avec l'affirmation à juger : « Le passage source dit bien ce que la citation lui fait dire. ». Marque en rouge les citations sous [0,9] et donne-moi la liste à vérifier. Avant, teste le montage sur [8] citations dont je connais la réponse, dont quelques fausses.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 04 23. Router tes fiches et tes e-mails dans n8n

Construis-moi une automatisation n8n avec le nœud TypeSafe AI. Pour chaque nouvelle demande arrivée par [formulaire / e-mail], utilise l'action Route pour l'envoyer vers le bon service parmi [devis, intervention, facturation, autre], chacun avec une phrase de description. Ajoute une sortie pour les demandes sous [0,6] de confiance, qui crée une tâche pour [personne] dans [mon CRM]. Explique-moi chaque nœud en une phrase.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 05 24. Extraire des dates d'un document

Pour chaque document de [dossier des conventions], découpe le texte utile autour de la date de fin. Demande à Jev, dans le même appel, trois choix : le jour (1 à 31), le mois (janvier à décembre) et l'année parmi [liste d'années], chacun avec une option « Introuvable ». Le code assemble la date, calcule le délai restant et prépare une relance [30] jours avant. Les documents où une réponse est sous [0,7] vont dans un dossier « à relire ».

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

PROMPT 06 25. Repérer les doublons entre deux catalogues

J'ai deux catalogues : [mon-catalogue.csv] et [catalogue-fournisseur.csv]. Dans le code, rapproche d'abord les paires probables (même catégorie, dimensions proches). Pour chaque paire, demande à Jev de juger vrai ou faux : « Ces deux lignes désignent le même produit. ». Donne-moi trois listes : doublons sûrs au-dessus de [0,9], à vérifier entre [0,5] et [0,9], nouveaux produits en dessous. Ne fusionne rien sans mon accord.

Si Jev n'est pas encore branché : utilise le skill TypeSafe et ma clé OPENROUTER_API_KEY (modèle typesafe/jev-1.13).

À FAIRE MAINTENANT

Choisis un seul cas parmi les 25, celui qui touche une tâche que tu fais chaque semaine. Colle sa consigne dans Claude Code, remplace les crochets, et fais tourner le résultat sur une dizaine de vrais exemples avant d'aller plus loin.

CHEZ SALES EXPERIENZ

Ce que Jev fait déjà dans mes propres machines

± 2 points

d'écart sur 100 en relançant 3 fois la note d'un même article (10 points avec un modèle rédacteur)

0,2 centime

d'euro pour classer 43 guides et publications LinkedIn en 6 thèmes

1 sur 3

des notes douteuses était fausse sur 45 entreprises notées : d'où la relecture des hésitations

Depuis le 30 septembre 2026, Jev ajoute une note de qualité « GEO » à chaque article de blog que publient mes machines n8n : est-ce que l'article sera cité par les IA ? Il juge 5 critères : va droit au but, citabilité, précision, confiance, titre aligné. Coût : environ 0,0001 dollar par article, soit environ un centième de centime d'euro. Le 5 octobre, j'ai fait noter 45 entreprises de l'Hérault par Jev (niveau d'IA, signaux d'achat) avant de les prospector : 0,0036 dollar au total, moins d'un centime d'euro. En relisant, une note douteuse sur trois était fausse. Depuis, je relis toutes les hésitations. Le 10 octobre, Claude a proposé une fois 6 thèmes, puis Jev a classé chacun de mes 43 contenus. Même chose pour mon dossier Téléchargements, 834

éléments et 7 Go, rangé par thème pour environ 4 centimes d'euro de Jev. Trois pièges vécus. Un nom de fichier vague donne un choix vague : Jev juge sur ce qu'on lui montre. Une catégorie fourre-tout aspire tout : 15 contenus sur 43 y sont tombés. Et Jev n'invente pas les catégories : c'est Claude qui les propose.

Bien s'en servir

Les quatre habitudes qui font marcher chaque consigne

01 Laisse une porte de sortie

Ajoute toujours une option « Autre » ou « Aucune ». Sans elle, Jev est obligé de choisir une mauvaise option quand rien ne colle. Décris chaque option en une phrase, et méfie-toi de la catégorie fourre-tout qui aspire tout.

02 Envoie les cas douteux à une personne

Fixe une ligne, par exemple 60 % de confiance, et tout ce qui passe dessous part chez quelqu'un. Garde le plancher à 0,5. Monte la ligne quand l'action engage, comme un devis ou une relance.

03 Teste sur 50 exemples dont tu connais la réponse

Avant de faire confiance, fais passer 50 cas déjà tranchés, en français, sans montrer les réponses à Jev. Compte les accords, regarde les erreurs, retouche les descriptions. Refais le test avec les options dans l'ordre inverse.

04 Jev décide, Claude écrit

Dès qu'une étape demande une vraie phrase, elle va chez Claude. Les calculs et les dates vont dans le code ou le tableur. Jev reste à sa place : trancher.

Les erreurs qui font dérailler un tri

Erreur n°1 Poser une question floue : Jev lit au pied de la lettre, écris la condition exacte

Erreur n°2 Lui demander de compter, de calculer ou de comparer deux dates

Erreur n°3 Lui envoyer tout le dossier client alors que trois lignes suffisent

Erreur n°4 Écrire des options qui se recoupent ou contredisent la consigne

Erreur n°5 Oublier l'option « Autre », puis s'étonner d'un mauvais choix forcé

Erreur n°6 Laisser Jev agir seul sur une décision qui engage, comme écarter un candidat ou envoyer un devis

Erreur n°7 Croire Claude Code quand il dit que tout marche, sans vérifier sur de vrais exemples

Erreur n°8 Coller la clé d'accès dans un fichier partagé ou dans une conversation

Un mot sur le marché.

OpenAI a lancé sa propre Decisions API avec le modèle Luna, présentée par la presse comme une réponse à Jev, et Microsoft Decision-1 est disponible sur la passerelle Vercel : apprendre à monter « un modèle qui décide + un modèle qui écrit » te servira, même si Jev n'est pas l'outil final.

À FAIRE MAINTENANT

Prends le cas choisi en partie 3. Rassemble 50 exemples dont tu connais la réponse, lance la consigne 8 dessus, et note le taux d'accord. Tant qu'il ne te convient pas, tu retouches les descriptions d'options, pas le seuil.

Questions fréquentes

Est-ce que Jev comprend le français ?

Oui, mais l'anglais reste sa langue la plus précise. TypeSafe dit lui-même que le français marche « mais pas aussi bien ». Teste-le sur 50 de tes propres exemples en français avant de lui confier un tri.

Combien ça coûte vraiment ?

0,042 dollar par million de jetons lus, environ 4 centimes d'euro, et les réponses sont gratuites. Mon test du 11 octobre 2026 : 397 jetons pour un message et deux questions, soit environ 60 000 décisions pour 1 euro. Les textes rédigés par Claude se paient à part.

Faut-il savoir coder ?

Non. Avec le chemin 1, tu colles des commandes dans le Terminal et tu donnes des consignes en français à Claude Code, qui écrit le programme. Tu dois en revanche savoir relire un plan et vérifier les résultats sur de vrais exemples.

Mes données ?

TypeSafe s'engage à ne pas entraîner ses modèles sur les demandes de ses clients. L'absence totale de conservation des données est réservée aux clients entreprise. N'envoie à Jev que le texte utile à la décision.

Jev remplace-t-il Claude ?

Non. Jev ne rédige rien et ne remplace pas le modèle de Claude Code. Claude écrit le programme et les réponses, Jev tranche à chaque passage.

Et s'il se trompe ?

Il ne sort jamais de ta liste, mais il peut choisir la mauvaise option. C'est pour ça que chaque réponse a une confiance, que les cas douteux vont à une personne, et que tu testes sur des exemples connus avant de lui faire confiance.

Sources, toutes lues le 11 octobre 2026.

Documentation de TypeScript : docs.typesafe.ai, docs.typesafe.ai/models, docs.typesafe.ai/introduction/quickstart, docs.typesafe.ai/introduction/coding-agents, docs.typesafe.ai/agent-skill, docs.typesafe.ai/confidence, la page « Jev 1.13 jaggedness » et les recettes « Guardrails for LLMs », « citation_check » et la carte des usages.

OpenRouter : openrouter.ai, son tutoriel Jev et openrouter.ai/docs/cookbook (« Auto-Approve Coding Agent Permission Prompts with Jev »). Vercel : vercel.com/docs/ai-gateway/getting-started/decision. n8n : n8n.io/integrations/typesafe-ai.

Presse et articles : it-connect.fr (25 septembre 2026), ayautomate.com (19 septembre 2026), thenewstack.io (fin septembre 2026), scalewise.com (1er octobre 2026). Catalogue de réalisations : shipwithjev.com.

Idées des cas 1 à 19 : le guide « Jev use cases that blow minds » de RoboNuggets.

Tests et chiffres de la partie « Chez Sales Experiencz » : mesures internes, du 30 septembre au 11 octobre 2026.

À télécharger

Toutes les consignes et commandes de ce guide existent aussi en fichier texte, prêtes à copier.

↓ [installation-jev.txt](#)

↓ [consignes-faciles.txt](#)

↓ [consignes-intermediaires.txt](#)

↓ [consignes-avancees.txt](#)

↓ [consignes-trouvees-en-ligne.txt](#)

↓ **Bonus : le skill « Ranger et classer avec Claude + Jev », à importer dans Claude (.zip)**

↓ **Le guide complet en PDF**

↓ **Tout le dossier en un seul fichier (.zip)**

ET MAINTENANT ?

Quelles décisions confier à Jev, lesquelles garder ?

En 45 minutes, on regarde ensemble les décisions répétitives de ton entreprise. On repère celles que Jev peut prendre en une fraction de seconde, et celles qui doivent rester humaines parce qu'elles engagent un client, un candidat ou ton chiffre d'affaires.

- ✓ La liste des décisions répétitives de ton entreprise qui peuvent passer chez Jev
- ✓ Celles qui doivent rester entre des mains humaines, et pourquoi
- ✓ Le chemin d'installation adapté à tes outils actuels
- ✓ Un premier cas à lancer cette semaine, avec ses questions et ses seuils
- ✓ Une estimation du temps gagné et du coût réel, en euros

Sans engagement. 100 % en visio.

Réserver mon échange de 45 minutes

salesexperienz.fr