Fiche mémo | Leçon 20 | 27 min

# Classer et extraire avec une sortie structurée en JSON

Une IA branchée dans un flux doit rendre des données que les étapes suivantes savent lire : un JSON imposé par un schéma, avec une catégorie prise dans une liste fermée, des montants en nombres, des dates au format AAAA-MM-JJ, null quand l'information manque, et un champ « incertain » qui envoie le cas douteux à une personne.

## Objectifs

- Écrire le schéma JSON d'une extraction, avec liste fermée, nombres, dates et valeurs nulles
- Régler une sortie structurée dans n8n, Make ou Zapier, ou par l'API d'OpenAI, Claude, Gemini et Mistral
- Tester un classement sur vingt cas réels et envoyer les cas douteux à une personne

## À retenir

- Dans un flux, l'IA rend un JSON imposé par un schéma, jamais une phrase.
- Le mode JSON garantit du JSON valide ; seule la sortie structurée garantit vos champs et leurs types.
- Liste fermée avec « autre », montants en nombres, dates en AAAA-MM-JJ, null quand l'information manque.
- Le champ « incertain » envoie le cas douteux à une personne : une erreur signalée ne coûte rien.
- Vingt messages réels, dont trois pièges, avant la mise en service et à chaque changement de modèle.

## Texte libre ou JSON imposé

**« Dis-moi de quoi il s'agit »**

- « Il s'agit d'une demande pour des fenêtres, probablement en bois »
- Une phrase différente à chaque appel
- Le routeur ne sait pas quoi lire
- Le montant arrive en texte, avec un espace et un symbole

**JSON imposé par un schéma**

- « categorie » : « fenetre », pris dans une liste fermée
- Les mêmes champs, dans le même ordre, à chaque appel
- Le routeur lit un seul champ
- « budget_eur » : 4500, un nombre que le tableur additionne

La même demande de devis, deux façons de faire répondre l'IA.

## Le code à recopier

**Le schéma JSON des demandes de devis (à coller tel quel)** JSON

```
{
  "type": "object",
  "properties": {
```

```
    "categorie": { "type": "string",
      "enum": ["fenetre", "porte", "escalier", "agencement", "reclamation", "autre"],
      "description": "Le sujet principal. autre si aucune catégorie ne convient." },
    "nom": { "type": ["string", "null"], "description": "Nom du client tel qu'écrit, null s'il
manque." },
    "telephone": { "type": ["string", "null"], "description": "Chiffres seuls, sans espaces, null
s'il manque." },
    "commune": { "type": ["string", "null"], "description": "Commune du chantier, null si absente."
},
    "materiau": { "type": "string",
      "enum": ["bois", "aluminium", "pvc", "mixte", "non_precise"] },
    "budget_eur": { "type": ["number", "null"],
      "description": "Budget annoncé en euros, nombre sans symbole ni espace. null si aucun
montant." },
    "date_souhaitee": { "type": ["string", "null"],
      "description": "Date au format AAAA-MM-JJ, seulement si un jour précis est écrit. Sinon null."
},
    "urgent": { "type": "boolean", "description": "true seulement si le client parle d'urgence ou de
sinistre." },
    "incertain": { "type": "boolean",
      "description": "true si un champ a été deviné, si le message a deux sujets ou s'il est
ambigu." },
    "resume": { "type": "string", "description": "Une phrase de 25 mots au plus, sans donnée
inventée." }
  },
  "required": ["categorie", "nom", "telephone", "commune", "materiau", "budget_eur",
               "date_souhaitee", "urgent", "incertain", "resume"],
  "additionalProperties": false
}
```

Claude refuse les contraintes « minimum » et « maxLength » dans un schéma envoyé tel quel : la longueur du résumé est donc donnée dans la description, pas en contrainte.

## Les erreurs fréquentes

| L'erreur, et pourquoi elle coûte | La correction |
|---|---|
| **Écrire « réponds uniquement en JSON » dans la consigne, ou choisir le mode JSON, et considérer que le format est garanti.**<br>Le JSON sera valide, mais un nom de champ peut changer, un champ manquer ou s'ajouter ; le flux casse une fois sur trente, souvent la nuit. | Régler une sortie structurée avec un schéma complet : tous les champs obligatoires, aucun champ en plus. |
| **Laisser les montants et les dates en texte libre (« 4 500 € », « 15 novembre »).**<br>Le tableur ne les additionne pas, ne les trie pas, et chaque étape suivante doit les nettoyer à sa façon. | Montant en nombre sans symbole, date en AAAA-MM-JJ, précisés dans le type et dans la description du champ. |
| **Donner une liste de catégories sans « autre », ou ne pas permettre null.**<br>L'IA est forcée de choisir une catégorie ou d'écrire une valeur : la candidature devient un devis, le budget absent devient 0. | Toujours une valeur « autre » routée vers une personne, et null permis pour chaque information qui peut manquer. |
| **Valider le classement sur trois exemples écrits par soi, puis le mettre en service.**<br>Vos exemples sont nets ; les vrais messages mélangent deux sujets, oublient l'essentiel ou n'ont rien à voir. | Vingt messages réels anonymisés, dont trois pièges, comptés dans une grille conservée pour les changements futurs. |