

Grok AI

Documentation — Manuel — FR

1 Introduction

Ce manuel présente *Grok* et son usage avec le Kit de documentation. Il s'adresse à quelqu'un qui veut produire des documents — guides, procédures, sites HTML — sans dépendre d'un seul fournisseur d'*intelligence artificielle*.

Le Kit est agnostique: chaque assistant a sa propre rubrique. Celle-ci décrit ce que *Grok* sait faire dans un projet isolé, comment il s'appuie sur les artefacts, et comment la chaîne de production rend un livrable homogène.

Note: *Les termes techniques sont définis dans le Glossaire du Kit. En cas de doute, consulter le glossaire avant de continuer.*

2 Identité

2.1 Produit et éditeur

Grok est un *grand modèle de langage* développé par *xAI*. Il lit et produit du texte en français, en anglais, en allemand et dans d'autres langues. Dans l'environnement projet, il exécute aussi des outils: lecture et écriture de fichiers, exécution de code, recherche web, *génératio*n de documents.

xAI publie plusieurs modèles en parallèle. Les noms changent. Les niveaux restent: rapide pour le volume simple, équilibré pour le travail courant, plus puissant pour le raisonnement long. Pour le Kit, rester au niveau de travail par défaut sauf si une tâche résiste.

2.2 Ce que Grok fait

- **Structurer.** Plans, sections, tableaux, flux de production.
- **Rédiger.** Texte technique clair, dans la langue du projet, au niveau demandé.
- **Exécuter.** Scripts **Node.js** et *Python*, validation, placement des fichiers dans les artefacts.
- **Vérifier.** Contrôles de *stylesheet*, niveaux, nommage, *empreinte* du .docx.
- **Reprendre.** Relire les *fichiers du projet* et continuer là où la session précédente s'est arrêtée.

2.3 Ce que Grok n'est pas

Grok n'est pas un moteur de recherche. Une réponse fluide n'est pas une preuve. Il n'a pas accès aux artefacts d'un autre projet. Il ne se souvient pas d'une *conversation* précédente, sauf par les fichiers déposés et, le cas échéant, la *mémoire persistante*.

Il peut halluciner — surtout sur des versions logicielles, des dates récentes et des chiffres. Pour tout contenu destiné à être publié ou utilisé en production, vérifier les faits critiques auprès des sources primaires.

💡 *Demander d'où vient une information est toujours légitime, y compris après une*

recherche web.

3 Environnement de travail

3.1 Projets isolés

Un projet *Grok* regroupe des conversations et des fichiers liés à un même domaine. Chaque projet est isolé: *Grok* ne voit pas les artefacts d'un autre projet. C'est voulu. Un *contexte* professionnel ne doit pas contaminer un *contexte* personnel, et un projet Kit ne doit pas mélanger ses règles avec celles d'un *projet consommateur*.

Ouvrir une *conversation* dans le bon projet. Une *conversation* hors projet n'a ni les fichiers de référence, ni les conventions de ce domaine.

3.2 Artefacts — source de vérité

Les fichiers présents dans les artefacts du projet courant sont la source de vérité. *Registry*, *Prompt*, feuilles de style, documents, glossaire, générateurs. La *mémoire vive* de la session disparaît. Les artefacts restent.

Élément	Rôle	Qui le maintient
Artefacts	Source de vérité entre sessions	Vous, après chaque livraison
Prompt projet	Règles permanentes de travail	Vous et <i>Grok</i> , sur validation
Registry	Versions, timestamps, langue, flags	Mis à jour à chaque livraison
Mémoire vive	Historique du <i>chat</i> en cours	Plateforme — non fiable seule

Note: *Après une mise à jour majeure, déposer la nouvelle version et retirer l'ancienne. Deux versions du même fichier produisent des règles contradictoires.*

3.3 Fenêtre de contexte

Tout ce qui s'écrit dans une *conversation* occupe la *fenêtre de contexte*: messages, réponses, fichiers joints. Quand la limite approche, le détail des premières instructions se dégrade. Pour un travail long, mieux vaut plusieurs sessions courtes, avec un résumé et des artefacts à jour, qu'une seule *conversation* interminable.



Si vous devez rappeler une règle déjà posée dans la même session, il est temps de clore et de reprendre proprement.

4 Grok et le Kit

4.1 Rôle dans la chaîne

Le Kit n'est pas un *modèle* de document Word à remplir à la main. C'est une chaîne: contenu validé, script de *génération*, *stylesheet*, *horodatage*

Luxembourg, validation, livraison. *Grok* orchestre cette chaîne dans le projet. Il n'invente pas la mise en forme: il appelle les fonctions de la *feuille de style* active.

#	Étape	Livrable
1	Cadrage — sujet, public, structure	Plan validé
2	Contenu — section par section	Texte validé
3	Script — appels <i>stylesheet</i> et <i>setters</i>	Générateur prêt
4	<i>Génération</i> — Packer + injectCustomProps	Document horodaté
5	Validation — kit_validate_docx et contrôles	Exit 0
6	Livraison — un fichier à la fois	<i>Artefact</i> à jour

Les contrôles amont et aval sont bloquants. Un échec interrompt la chaîne: rien n'est livré tant qu'elle n'a pas entièrement passé.

4.2 Capacités utiles pour la documentation

- **Lecture intégrale.** *Prompt*, Reference *stylesheet*, *Registry* — avant toute *génération*.
- **Niveaux stricts.** p2 sous un h2, note3 sous un h3. Pas de formatage manuel hors helpers.
- **Nommage.** Préfixe - Catégorie - Sujet (AAAA-MM-JJ - HHhMM).ext, *horodatage* via getLuxTimestamp().
- **Publication.** Une fois les .docx validés, le générateur de site produit HTML, CSS, PDF et glossaire.
- **Reprise.** Relire les artefacts, résumer l'état, annoncer le plan, attendre le feu vert.

4.3 Ce que le Kit impose à Grok

Annoncer avant d'agir. Attendre confirmation avant de modifier un document existant. Préserver le contenu source. Livrer de façon séquentielle. Ne jamais reconstruire un document à partir de la seule mémoire de session.

💡 *Si une règle semble oubliée en milieu de longue session, un rappel court suffit. C'est aussi le signe que la fenêtre de contexte se remplit.*

5 Autres projets de documentation

5.1 Injecter le Kit, pas le bricoler

Pour un nouveau domaine — domotique, réseau, collection, procédure familiale — on copie le Kit complet dans le *projet consommateur*. Pas de

cherry-picking. Les feuilles de style, le *Registry*, les *validateurs* et les conventions voyagent ensemble.

Grok initialise ensuite le préfixe, la langue, l'auteur et l'*inventaire* selon le *Guide d'initialisation*. Le résultat: les mêmes réflexes, le même rendu, un site distinct.

5.2 Ce qui change d'un projet à l'autre

- **Le préfixe.** Jamais “Kit” hors adaptation du Kit lui-même.
- **La langue.** `project.docLanguage` dans le *Registry*. Le Kit est en FR par exception; les consommateurs démarrent en EN.
- **Les modules data.** Glossaire, marques, éventuellement variables et entités — selon `Registry.requires`.
- **Le contenu.** C'est le seul endroit où le projet s'exprime. Le moteur de formatage reste celui du Kit.

Note: *Une évolution de stylesheet se fait dans le projet Kit, puis s'injecte en bloc dans les consommateurs. Voir le Guide de mise à jour.*

6 Limites et vérifications

6.1 Faits, versions, dates

Les erreurs portent surtout sur les données précises. Une recherche web réduit le risque; elle ne le supprime pas. Deux sources peuvent se contredire. Pour une procédure, une version logicielle ou un chiffre publié, revenir à la documentation officielle.

6.2 Fichiers et forme

Joindre un `.docx` aux artefacts n'en conserve pas la mise en forme: la plateforme en extrait le texte. Pour modifier un document existant, déposer le binaire dans la *conversation* en cours et régénérer via le *stylesheet* — jamais patcher à l'aveugle le XML si un helper Kit existe.

Note: *Ne pas compter sur une continuité entre projets. Ce qui a été établi dans l'un doit être réintroduit dans l'autre, par fichiers ou par énoncé explicite.*

7 Suite

Ce manuel fixe le cadre. Le Guide pratique *Grok* donne les réflexes de session, les techniques de formulation, les flux par type de travail et les gabarits prêts à envoyer. Le Kit évolue. Les principes de ce document restent: artefacts comme source de vérité, *génération* via *stylesheet*, validation avant livraison, isolation des projets.