Eigenflow

GUIDE GRATUIT · LINKEDIN

# LinkedIn Agent Skill : 11 skills Claude pour ton LinkedIn

Installer, écrire, et ce qui coince en français : notre test du 31/10


PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb



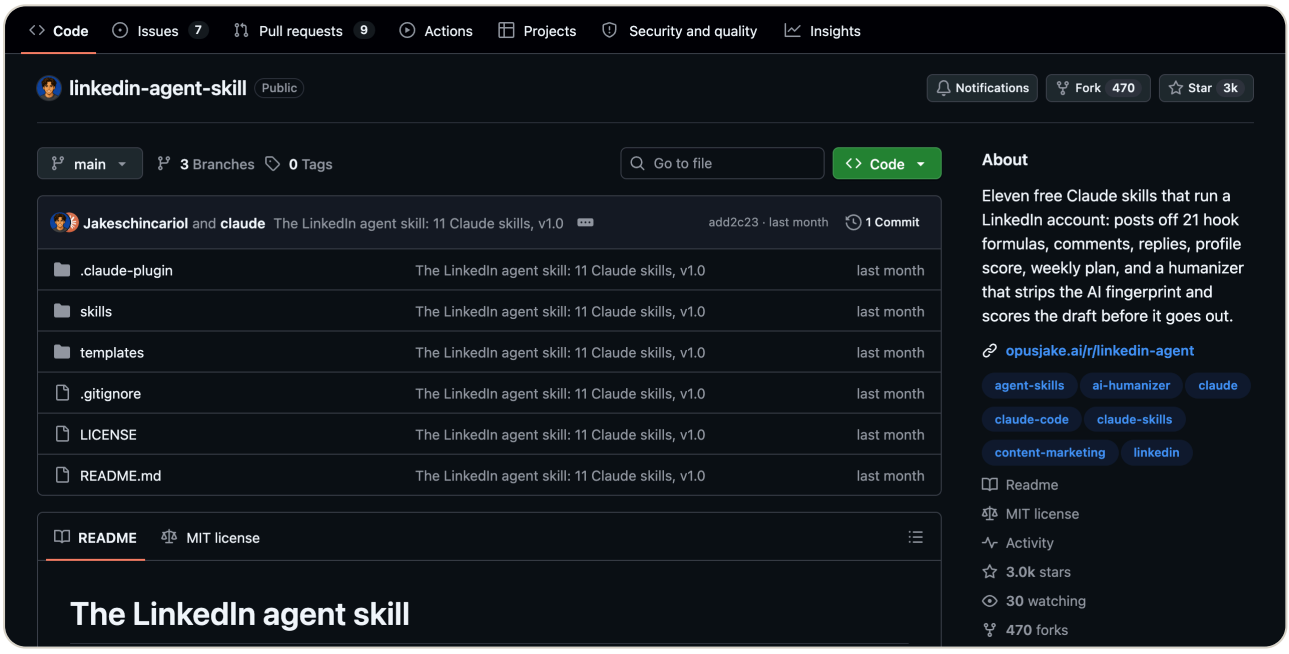
ÉDITION

octobre 2026

Tu as commenté SKILL. Ce guide explique comment installer LinkedIn Agent Skill dans Claude, comment t'en servir, et ce qu'il faut savoir si tu écris en français. Tout vient du dépôt tel qu'il était le 31 octobre 2026 (commit add2c23) et de notre test du même jour, lancé dans une session Claude Code neuve, sur notre propre histoire. Aucun compte LinkedIn n'a été touché.

## Ce que c'est

Un dossier de onze skills pour Claude, écrit par Jake Schincariol. Le dépôt s'appelle Jakeschincariol/linkedin-agent-skill, sous licence MIT : tu peux l'utiliser, le modifier et le revendre, en gardant la mention de copyright. Le 31 octobre, il avait 2 967 étoiles et 470 forks.




La page du dépôt le 31 octobre.

Deux choses à savoir avant de commencer, écrites par l'auteur lui-même dans le README :

- **Les skills écrivent, c'est toi qui publies.** Rien ne part sur LinkedIn tout seul. LinkedIn n'offre pas d'API pour publier sur un profil personnel sans application partenaire, et automatiser le site avec un navigateur enfreint ses conditions d'utilisation. Chaque skill finit donc par un bloc à copier-coller.

- **Les cinq tests du humaniseur tournent sur ta machine.** Ce ne sont pas GPTZero, Originality ou Turnitin, et ils ne promettent aucun verdict de ces outils.

## Les onze commandes

- **/li-post** : une idée devient un post : trois accroches tirées de 21 formules, un brouillon complet, humanisé avant que tu le voies.

- **/li-comment** : des commentaires sur les posts des autres, neuf types choisis selon le post. Jamais « Great post! ».

- **/li-reply** : les réponses sous tes posts : il trie les commentaires (prospect, fond, pair, soutien, bruit) et répond dans cet ordre.

- **/li-profile** : note ton profil sur 100 avec une grille de 12 critères, puis le réécrit en commençant par ce qui coûte le plus de points.

- **/li-plan** : la semaine : quoi poster, quand, et les 10 personnes avec qui échanger.

- **/li-human** : le humaniseur : deux scripts qui nettoient et notent un texte.

- **/li-carousel** : le texte d'un carrousel slide par slide, la couverture, et le PDF à téléverser.

- **/li-repurpose** : une vidéo, une newsletter ou une transcription devient une semaine de posts.

- **/li-dm** : la note d'invitation de 200 caractères, le premier message et deux relances.

- **/li-inbox** : trie ta messagerie : prospect, recruteur, pair, demande, spam.

- **/li-audit** : l'analyse de ce que tu as déjà publié, classé par taux d'engagement.

---

## Étape 1 · Installer

Le plus simple : colle ces deux lignes dans Claude.

**À COLLER**

```
https://github.com/Jakeschincariol/linkedin-agent-skill

Install this skill, then confirm /li-post works.
```

Dans Claude Code, tu peux aussi les installer à la main :

**À COLLER**

```
git clone https://github.com/Jakeschincariol/linkedin-agent-skill.git
cp -r linkedin-agent-skill/skills/li-* ~/.claude/skills/
```

Ou comme plugin :

À COLLER

```
/plugin marketplace add Jakeschincariol/linkedin-agent-skill
/plugin install linkedin-agent
```

Sans Claude Code, tu peux coller le fichier SKILL.md d'une seule skill en haut d'une conversation : elle marche comme un mode, mais tu perds les deux scripts Python du humaniseur.

---

## Étape 2 · Dix minutes sur ta voix

Toutes les skills lisent un fichier `~/.claude/linkedin/voice.md`. Copie le modèle `templates/voice.md` du dépôt à cet endroit et remplis-le, ou colle trois de tes posts dans Claude en lui demandant : « write my voice.md from these ». Sans ce fichier, les posts sonnent comme ceux de tout le monde. Dans notre test, la session tournait en mode restreint et ne pouvait pas le lire : le post a pris le ton de la demande.

---

## Étape 3 · Écrire un post

Tape `/li-post` suivi de ton idée, avec les vrais chiffres. Notre test, sur l'histoire de notre audit de sécurité du 30 octobre : en 1 minute 36, la skill a proposé trois accroches, chacune tirée d'une formule nommée, en a choisi une et a écrit le post complet, d'environ 1 000 caractères.

```
/li-post · session Claude Code neuve · 1 min 36

Accroches
1. #3 Aveu d'erreur : Notre API répondait à n'importe qui.
   Sans mot de passe.
2. #10 Les chiffres : 18 agents, 23 minutes, 8 pistes.
   La première était la bonne.
3. #19 Écart de curiosité : Le meilleur audit de sécurité
   qu'on ait lancé n'a rien confirmé tout seul.

J'ai choisi la #3 : c'est la faille elle-même qui arrête le scroll.
```

Les trois accroches proposées par /li-post, dans notre test du 31 octobre.

Si ton idée a besoin d'un chiffre que tu n'as pas donné, la skill le laisse en trou, `{{your number}}`, et te le signale. Elle n'invente pas de résultats à ta place.

---

## Le humaniseur en français : ce qui coince

Le humaniseur fait trois passes : il retire les caractères invisibles, corrige la typographie (tiret cadratin, guillemets) et remplace 113 tournures typiques de l'IA. Puis il note le texte sur cinq tests. Il a été écrit pour l'anglais, et ça se voit sur un texte français.

**Il retire les espaces avant les deux-points.** Sur notre post, « vraie : » est devenu « vraie: » et « API ? » est devenu « API? ». La règle est à la ligne 147 de `skills/li-human/humanize.py` :

**À COLLER**

```
text = re.sub(r"\s+([,.;:!?])", r"\1", text)
```

Pour garder la typographie française, remplace `[,.;:!?]` par `[,.]` sur cette ligne. Les espaces avant la virgule et le point continuent de partir, ceux avant `: ; ! ?` restent.

```
HUMANIZE REPORT  -  draft.txt
3 machine artefacts removed, 0 structural tells flagged

2. TYPOGRAPHY
    3x  →  ->   (RIGHTWARDS ARROW)

et, sans le dire : « vraie : » devient « vraie: », « API ? » devient « API? »
```

Le rapport du humaniseur sur notre post.

**Sa note reste bloquée sous 60.** Le test VOICE ne compte que les contractions et les pronoms anglais (don't, it's, I, we, you). Sur un texte français, il reste à 30, et il tire la note globale vers le bas : notre post a eu 59,4 sur 100, verdict REVIEW, avant comme après le nettoyage. Le test du lexique, lui, cherche des mots anglais : son 100 sur 100 ne prouve rien en français.

```
AI DETECTION PANEL  -  draft.txt  (notre post en français, 31/10)

  BURSTINESS    ####################.....  80.8
  SPECIFICITY   ####################....  84.1
  SLOP DENSITY  ######################### 100.0
  FINGERPRINT   ######################### 100.0
  VOICE         #######.................  30.0
                0.0 contractions, 0.0 personal pronouns per 100 words
------------------------------------------------------------
  HUMAN SCORE   ##############..........  59.4   REVIEW

  59.4 REVIEW  ->  59.4 REVIEW   (+0.0)
```

Le panneau de notes sur notre post en français.

**Il ne repère pas les tournures françaises.** Le post écrit par `/li-post` gardait une phrase en « c'est X, pas Y », que le humaniseur signale en anglais mais pas en français.

## Ce qu'on te conseille en français

1. Remplis ton `voice.md` avec trois de tes vrais posts.

2. Utilise les skills d'écriture : `/li-post`, `/li-comment`, `/li-reply`, `/li-plan`, `/li-dm`, `/li-carousel`. Elles marchent bien en français.

3. Corrige la ligne 147 de `humanize.py` avant de passer un texte français dans `/li-human`, et ne te fie pas à sa note.

4. Relis toi-même avant de publier : cherche les tirets cadratins, les phrases en « ce n'est pas X, c'est Y », les listes de trois, les questions d'appât en fin de post.

Il existe une version espagnole du dépôt (aldushmedia/linkedin-agent-skill-es). Il n'y a pas encore de version française.

---

## Les liens

- Le dépôt : https://github.com/Jakeschincariol/linkedin-agent-skill

- L'article de l'auteur : https://opusjake.ai/r/linkedin-agent