

Skille globalne, lokalne, Windows/WSL i screenshoty w Claude Code CLI

Krótki FAQ dla osób, które po szkoleniu chcą zainstalować skille, używać ich w różnych projektach i wrzucać agentowi screenshoty albo obrazki.

1. Najkrócej

- **Skill globalny/osobisty** działa we wszystkich projektach danego użytkownika.
 - **Skill lokalny/projektowy** działa tylko w konkretnym projekcie lub repozytorium.
 - **Claude Code czyta skille automatycznie**, jeśli są w odpowiednim folderze.
 - **Codex nie musi widzieć skilli Claude Code**. Traktuj je wtedy jako instrukcje/prompt do skopiowania albo przenieś logikę do `AGENTS.md`.
 - **Screenshoty w Claude Code działają**: przez przeciągnięcie pliku, `Ctrl+V` z obrazem albo podanie ścieżki do pliku.
 - Na Windows/WSL najpewniejszy workflow to: **zapisz screenshot jako plik i podaj agentowi ścieżkę**.
-
-

2. Co to jest skill

Skill to mała procedura dla agenta: instrukcja, kiedy i jak ma wykonać konkretny typ zadania.

Przykład:

```
6-pole-brief/  
SKILL.md
```

W `SKILL.md` są dwie części:

```
---  
description: Use when the user wants to turn a vague task into a clear agent brief.  
---
```

Instrukcja

Zadaj użytkownikowi pytania o cel, kontekst, źródła, ograniczenia, jakość i definicję gotowe.

Nazwa folderu staje się nazwą skilla. Jeśli folder nazywa się `6-pole-brief`, możesz go wywołać jako:

```
/6-pole-brief
```

albo napisać naturalnie:

```
Pomóż mi zbudować dobry brief dla agenta.
```

Claude może wtedy sam zdecydować, że ten skill pasuje.

3. Skill globalny / osobisty

Skill globalny jest dostępny w każdym projekcie, w którym uruchamiasz Claude Code jako ten sam użytkownik.

Ścieżka na macOS/Linux/WSL:

```
~/.claude/skills/nazwa-skilli/SKILL.md
```

Przykład:

```
mkdir -p ~/.claude/skills/6-pole-brief  
cp SKILL.md ~/.claude/skills/6-pole-brief/SKILL.md
```

Kiedy używać globalnie:

- masz procedurę, której używasz wszędzie,
- to jest twój osobisty sposób pracy,
- nie chcesz kopiować skilla do każdego projektu.

Przykłady dobrych globalnych skilli:

- budowanie briefu,
 - grillowanie pomysłu,
 - review planu,
 - przygotowanie odpowiedzi w twoim stylu,
 - rutyna bezpieczeństwa przed odpaleniem agenta.
-
-

4. Skill lokalny / projektowy

Skill lokalny siedzi w projekcie i jest dostępny tylko tam.

Najczęstsza ścieżka:

```
.projekt/.claude/skills/nazwa-skilla/SKILL.md
```

Przykład w repozytorium:

```
moj-projekt/  
  .claude/  
    skills/  
      release-checklist/  
        SKILL.md  
src/  
README.md
```

Kiedy używać lokalnie:

- skill dotyczy konkretnego projektu,
- zawiera nazwy plików, komend, środowisk albo procesów z tego repo,
- chcesz go wersjonować razem z projektem,
- kilka osób w zespole ma korzystać z tej samej procedury.

Przykłady lokalnych skilli:

- `deploy-checklist`,
 - `client-report`,
 - `n8n-workflow-review`,
 - `safe-folder-audit`,
 - `test-this-app`.
-

5. Który wariant wybrać

Jeśli skill opisuje **twój styl pracy** albo uniwersalną procedurę, daj go globalnie.

Jeśli skill opisuje **ten konkretny projekt**, daj go lokalnie.

Prosty test:

Czy ten skill ma sens poza tym projektem?

- Tak → globalny.
- Nie → lokalny.

Drugi test:

Czy w skillu są nazwy plików, endpointów, klientów, komend albo tajniki projektu?

- Tak → lokalny.
 - Nie → może być globalny.
-

6. Czy skille zjadają tokeny

Trochę tak, ale nie tak jak długi prompt wklejany za każdym razem.

W Claude Code:

- lista skilli i ich opisy pomagają modelowi zdecydować, czego użyć,
- pełna treść `SKILL.md` ładuje się dopiero, gdy skill jest użyty,
- dlatego dłuższy skill zwykle kosztuje mniej niż wklejanie całej instrukcji do każdej rozmowy.

Praktyczna zasada:

- `description` ma być krótki i konkretny,
 - długa instrukcja może być w body skilla,
 - duże materiały referencyjne trzymaj jako osobne pliki obok skilla i każ agentowi czytać je tylko wtedy, gdy są potrzebne.
-

7. Czy Codex widzi skille Claude Code

Nie zakładaj tego.

Claude Code ma własny mechanizm skilli w `.claude/skills/` i `~/.claude/skills/`. Inne narzędzia agentowe mogą mieć własne formaty instrukcji.

Dla Codex / innych agentów najbezpieczniejsze warianty są trzy:

Wariant A: skopiuj treść skilla jako prompt

Otwórz `SKILL.md`, skopiuj instrukcję i wklej ją do rozmowy z Codexem.

Wariant B: trzymaj instrukcję w `AGENTS.md`

Jeśli pracujesz w repo, dodaj ogólne zasady projektu do:

```
AGENTS.md
```

To jest lepsze dla zasad projektu niż dla długich procedur krok po kroku.

Wariant C: powiedz agentowi, który plik ma przeczytać

Przykład:

```
Przeczytaj .claude/skills/safe-folder-audit/SKILL.md i wykonaj tę procedurę dla tego folderu.
```

To działa jako zwykła instrukcja z pliku, nawet jeśli narzędzie nie obsługuje skilli natywnie.

8. Windows, WSL i PowerShell

Najważniejsze: Windows i WSL to dla narzędzi dwa różne światy plików.

Jeśli uruchamiasz Claude Code w WSL

Globalne skille trzymaj tutaj:

```
~/ .claude/skills/
```

czyli w praktyce np.:

```
/home/twoj-user/.claude/skills/
```

Projekt najlepiej trzymaj też po stronie WSL, np.:

```
/home/twoj-user/projekty/moj-projekt
```

A nie na:

```
/mnt/c/Users/...
```

Dlaczego? Bo praca na `/mnt/c/...` bywa wolniejsza, potrafi inaczej obsługiwać uprawnienia i czasem robi dziwne rzeczy z watcherami plików.

Jeśli plik jest po stronie Windows

Windowsowy plik:

```
C:\Users\Anna\Pictures\screenshot.png
```

w WSL będzie zwykle widoczny jako:

```
/mnt/c/Users/Anna/Pictures/screenshot.png
```

Czyli do Claude Code w WSL piszesz:

```
Przeanalizuj obraz: /mnt/c/Users/Anna/Pictures/screenshot.png
```

Jeśli ścieżka ma spację, użyj cudzysłowu:

```
Przeanalizuj obraz: "/mnt/c/Users/Anna/Pictures/Moje Screeny/error 01.png"
```

Jeśli uruchamiasz Claude Code w PowerShell

Ścieżki będą windowsowe:

```
mkdir $HOME\.claude\skills\6-pole-brief  
copy .\SKILL.md $HOME\.claude\skills\6-pole-brief\SKILL.md
```

W promptach podawaj ścieżki w cudzysłowie, jeśli mają spacje:

```
Przeanalizuj obraz: "C:\Users\Anna\Pictures\error screen.png"
```

9. Jak sprawdzić, czy skill jest widoczny

Po instalacji uruchom Claude Code w projekcie i wpisz:

```
/skills
```

albo spróbuj bezpośrednio:

```
/6-pole-brief
```

Jeśli skill nie pojawia się:

1. Sprawdź nazwę folderu.
2. Sprawdź, czy plik nazywa się dokładnie `SKILL.md`.
3. Sprawdź, czy frontmatter zaczyna się i kończy `---`.
4. Uruchom Claude Code ponownie.
5. Jeśli to skill lokalny, upewnij się, że jesteś w dobrym folderze projektu.

10. Screenshoty i obrazy w Claude Code CLI

Claude Code potrafi pracować z obrazami. Oficjalnie wspierane są trzy wygodne metody:

1. **Przeciągnij obraz do okna Claude Code.**

2. **Skopiuj obraz i wklej do CLI przez `ctrl+v`** . Ważne: w dokumentacji Anthropic jest wskazane `Ctrl+V` , nie `Cmd+V` .

3. **Podaj ścieżkę do pliku obrazu.**

Najpewniejsza metoda, szczególnie na Windows/WSL:

```
Zapisz screenshot jako plik i podaj agentowi ścieżkę.
```

Przykład:

```
Przeanalizuj ten screenshot i powiedz, co jest nie tak z layoutem: ./screenshots/layout-error
```

Albo:

```
To jest screen błędu z aplikacji. Znajdź prawdopodobną przyczynę i zaproponuj pierwszy bezpie
```

11. Proponowany workflow na screenshoty

W projekcie utwórz folder:

```
mkdir -p screenshots
```

Zapisuj pliki z prostymi nazwami:

```
screenshots/2026-07-09-login-error.png  
screenshots/2026-07-09-dashboard-before.png  
screenshots/2026-07-09-dashboard-after.png
```

Potem pytasz agenta:

```
Porównaj te dwa screenshoty i wskaż różnice w UI:
```

```
./screenshots/2026-07-09-dashboard-before.png  
./screenshots/2026-07-09-dashboard-after.png
```

Albo:

```
Na podstawie tego screenshotu zaproponuj CSS/HTML dla podobnego komponentu. Nie edytuj jeszcze  
./screenshots/component-mockup.png
```

12. Screenshoty na Windows

Najprościej

1. Naciśnij `Win + Shift + S`.
2. Zaznacz fragment ekranu.
3. Jeśli Claude Code przyjmuje obraz ze schowka, wklej `Ctrl+V`.
4. Jeśli nie, zapisz obraz jako plik i podaj ścieżkę.

Najbezpieczniejszy wariant:

```
C:\Users\Anna\Pictures\Screenshots\error.png
```

W WSL ta sama ścieżka będzie zwykle:

```
/mnt/c/Users/Anna/Pictures/Screenshots/error.png
```

Prompt:

```
Przeanalizuj screenshot: /mnt/c/Users/Anna/Pictures/Screenshots/error.png
```

13. Screenshoty na macOS

Typowe skróty:

- `Cmd + Shift + 4` - zaznaczenie fragmentu ekranu,
- `Cmd + Shift + 5` - panel screenshotów i nagrywania,

- `Ctrl` dodany do skrótów zwykle kopiuje do schowka zamiast zapisywać plik.

W Claude Code najpewniejsze opcje:

- przeciągnij zapisany plik do terminala,
- albo podaj ścieżkę,
- albo wklej obraz przez `Ctrl+V`, jeśli terminal to obsługuje.

Uwaga: `Cmd+V` w terminalu często oznacza zwykłe wklejanie tekstu przez aplikację terminala. Dokumentacja Claude Code wskazuje `Ctrl+V` dla wklejania obrazu do CLI.

14. Screenshotsy na Linux

Zależnie od środowiska możesz użyć np. narzędzia systemowego, Flameshot, Spectacle albo GNOME Screenshot.

Najstabilniejszy workflow:

```
mkdir -p screenshots
```

Zapisz plik do projektu, np.:

```
screenshots/error.png
```

I podaj agentowi:

```
Przeanalizuj screenshot: ./screenshots/error.png
```

Jeśli wklejanie ze schowka nie działa, to zwykle problem terminala/schowka, nie samego Claude Code. Użyj ścieżki do pliku.

15. Co mówić agentowi przy obrazach

Nie dawaj samego obrazka bez kontekstu. Powiedz, czego szukasz.

Słabo:

Co tu widzisz?

Lepiej:

To screenshot błędu po kliknięciu „Zapisz”. Powiedz:

1. co jest widoczne,
2. jaka może być przyczyna,
3. jaki jeden bezpieczny test mam zrobić najpierw.

Do UI:

To mockup ekranu dashboardu. Opisz strukturę komponentów i zaproponuj HTML/CSS. Nie implementuj

Do diagramu:

To diagram architektury. Wypisz komponenty, przepływy danych i miejsca, gdzie brakuje informac

Do błędu:

To screenshot terminala z błędem. Przepisz istotny komunikat, wyjaśnij go prostym językiem i

16. Typowe problemy

Skill nie działa

Najczęstsze przyczyny:

- zły folder,
- plik nie nazywa się `SKILL.md`,
- brak frontmatter `---`,
- Claude Code był uruchomiony zanim folder skilla powstał,
- jesteś w innym środowisku: Windows vs WSL.

Rozwiązanie:

```
Sprawdź ścieżkę, uruchom Claude Code ponownie, wpisz /skills.
```

Codex nie widzi skilla

To normalne. Skille Claude Code nie są automatycznie wspólnym standardem wykonania dla każdego narzędzia.

Rozwiązanie:

```
Wklej treść SKILL.md jako prompt albo każ Codexowi przeczytać ten plik.
```

Screenshot nie wkleja się do CLI

Najczęstsze przyczyny:

- terminal nie przekazuje obrazu ze schowka,
- używasz złego skrótu,
- obraz jest w schowku Windows, a Claude Code działa w WSL,
- sesja terminala jest zdalna przez SSH/tmux i nie ma dostępu do lokalnego schowka.

Rozwiązanie:

```
Zapisz obraz jako plik i podaj ścieżkę.
```

Agent nie widzi pliku ze screena

Sprawdź, czy ścieżka istnieje z perspektywy tego środowiska, gdzie działa Claude Code.

W WSL:

```
ls -lh /mnt/c/Users/Anna/Pictures/Screenshots/error.png
```

W PowerShell:

```
Get-Item "C:\Users\Anna\Pictures\Screenshots\error.png"
```

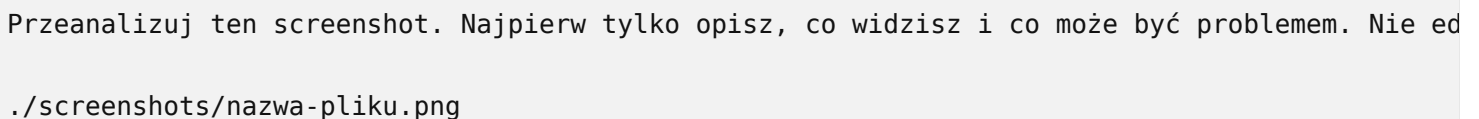
17. Bezpieczna zasada na start

Dla początkujących najlepszy zestaw to:

1. Skille osobiste instaluj w `~/.claude/skills/`.
2. Skille projektowe trzymaj w `.claude/skills/` w repozytorium.
3. Na Windows/WSL zapisuj screenshoty jako pliki.
4. Do agenta podawaj ścieżkę i konkretną prośbę.
5. Przy analizie obrazów zaczynaj od trybu read-only: „nie edytuj jeszcze plików, najpierw opisz plan”.

Gotowy prompt:

```
Przeanalizuj ten screenshot. Najpierw tylko opisz, co widzisz i co może być problemem. Nie edytuj plików.  
./screenshots/nazwa-pliku.png
```



18. Mini-checklista przed publikacją własnego skilla

- Nazwa folderu jest krótka i zrozumiała.
- Plik nazywa się dokładnie `SKILL.md`.
- `description` mówi, kiedy używać skilla.
- Instrukcja ma jasne kroki.
- Skill nie zawiera sekretów, tokenów, prywatnych URL-i ani danych klientów.
- Skill ma domyślnie bezpieczne zasady: nie kasuj, nie nadpisuj, nie wysyłaj bez zgody.
- Przetestowano go w nowej sesji Claude Code.