

- Krok 1. Ustaw tryb zatwierdzania
- Krok 2. Ustaw reguły dla niebezpiecznych komend
- Krok 3. Uruchom ponownie OMP i sprawdź
- Jak to działa
- Następny krok

Domyślnie OMP działa w trybie **yo!o**: sam zatwierdza każde narzędzie, także komendy w terminalu. Ten krok jest wymagany przed pracą na prawdziwym projekcie.

Czas: 3 minuty. Wymagania: **1. OMP: instalacja i podłączenie**. Efekt: OMP pyta o zgodę przed każdą komendą w terminalu i blokuje komendy niebezpieczne.

Krok 1. Ustaw tryb zatwierdzania

Linux i Windows

```
omp config set tools.approvalMode write
```

Krok 2. Ustaw reguły dla niebezpiecznych komend

Linux: uruchom komendę.

Linux

```
omp config set bash.patterns '[{"match":"git  
push*","approval":"prompt"}, {"match":"git reset  
--hard*","approval":"prompt"}, {"match":"git  
clean*","approval":"prompt"}, {"match":"rm  
-rf*","approval":"prompt"}, {"match":"* |  
sh","approval":"deny"}]'
```

Windows: PowerShell 5.1 gubi cudzysłowy " w tej komendzie. Zamiast niej dopisz poniższy blok do pliku %USERPROFILE%\omp\agent\config.yml. Na Linuksie ten sam blok w ~/omp/agent/config.yml daje ten sam efekt co kroki 1 i 2.

```
config.yml

tools:
  approvalMode: write
bash:
  patterns:
    - match: git push*
      approval: prompt
    - match: git reset --hard*
      approval: prompt
    - match: git clean*
      approval: prompt
    - match: rm -rf*
      approval: prompt
    - match: "*" | sh"
      approval: deny
```

Krok 3. Uruchom ponownie OMP i sprawdź

```
Linux i Windows

omp config get tools.approvalMode
```

Oczekiwany wynik: write. Przy następnej komendzie w terminalu OMP zapyta o zgodę (Approve / Deny).

Jak to działa

Tryby (tools.approvalMode)

Tryb	Bez pytania	Pyta o zgodę
always-ask	tylko odczyt plików	zapis plików i komendy
write (zalecany)	odczyt i zapis plików w projekcie	komendy w terminalu, kod (eval), przeglądarka, subagenci
yo!o (domyślny)	wszystko	tylko wzorce z bash.patterns i komendy krytyczne

Niezależnie od trybu OMP zawsze pyta przy komendach krytycznych, m.in. `rm -rf /`, `sudo rm`, `mkfs`, `dd ... of=/dev/...`, `curl ... | sh`, `shutdown`.

Wzorce komend (**bash.patterns**)

Pole	Wartość
match	Cała komenda, * oznacza dowolny ciąg znaków. <code>git push*</code> pasuje do <code>git push origin main</code> , nie pasuje do <code>git status</code> .
approval	prompt (zapytaj), deny (zablokuj), allow (wykonaj bez pytania).

- Wygrywa pierwszy pasujący wzorec, więc `deny` wpisuj przed `allow`.
- `omp config set bash.patterns` nadpisuje całą listę. Za każdym razem podawaj komplet wzorców.

Przykład: testy bez pytania, `git push` z pytaniem:

```
Linux
omp config set bash.patterns ' [{"match": "git push*", "approval": "prompt"}, {"match": "npm test*", "approval": "allow"} ] '
```

Tryb na jedną sesję

```
Linux i Windows
omp --approval-mode always-ask
```

Nie uruchamiaj `omp --auto-approve`. Ta flaga wyłącza wszystkie pytania o zgodę.

Następny krok

6. Pamięć długoterminowa (Hindsight)