

/goal – как заставить Claude работать пока ты спишь

Что такое /goal

- Ты пишешь `/goal` и одну финишную линию. Claude начинает работать немедленно — без второго промпта.
- Финишная линия = описание состояния, которое Claude может проверить сам.
- Claude работает шаг за шагом пока не достигнет условия. Ты не касаешься клавиатуры.

Как это работает



Worker

Основная модель (Opus / Sonnet). Делает работу. Пишет код, создаёт файлы, запускает команды.



Evaluator

Маленькая быстрая модель (Haiku). После каждого шага отвечает на один вопрос: «условие выполнено?»

- Если нет — говорит worker'у почему. Worker идёт дальше.
- Если да — goal закрывается сам. Контроль возвращается тебе.
- Evaluator — не та же модель. Она не пишет код.
- Пока goal активен — видишь индикатор `🕒 /goal active` с таймером.

Как писать условие правильно

⚠ Это — самый важный раздел. Если облажаться здесь — сожжёшь токены впустую.

Плохие условия — Claude крутится в петле

✗ `сделай красиво` — как evaluator проверит «красиво»?

✗ `сделай без ошибок` — что значит «без ошибок»?

✗ `организуй мои файлы` — когда «организовано»?

⚠ Один разработчик с таким условием утром обнаружил что сжёг весь недельный лимит — \$200 за ночь.

Хорошие условия — Claude останавливается сам

✓ `Every file in receipts/ renamed with date\_vendor format, sorted into monthly folders, spending CSV exists with one row per receipt`

✓ `rg 'legacy\_client' returns no matches under services/checkout; go test ./... exits 0`

✓ `CHANGELOG.md has an entry for every PR merged this week`

Три компонента хорошего условия

1. Одно измеримое состояние — результат теста, количество файлов, выход команды.
2. Способ проверки — как именно Claude докажет это в разговоре.
3. Ограничения — что нельзя менять по пути.

or stop after 25 turns — это **твой ремень безопасности**.

Автоподтверждение и настройка

По умолчанию Claude спрашивает разрешения на каждое действие. Если `/goal` активен — он всё равно будет останавливаться ждать твоего «да».

 Чтобы Claude работал полностью без тебя — включи auto-approve в настройках.

 Ты даёшь AI разрешение работать самостоятельно. Это нормально — если ты понимаешь к чему у него есть доступ. Claude не запустит ничего опасного без предупреждения. Но всё равно — включай когда понимаешь что делаешь.

Три правила перед включением auto-approve



Проверь `/usage`

Посмотри ситуацию с токенами



Добавь safety cap

``or stop after N turns`` в условие



Начни с малого

Не с «реорганизуй весь проект» — с маленькой задачи