

Руководство по синхронизации Claude Code

Как держать настройки Claude Code одинаковыми на всех компьютерах

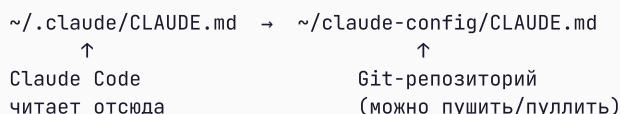
Что такое симлинки?

Симлинк (символическая ссылка) — это как ярлык на рабочем столе, только для файлов в терминале.



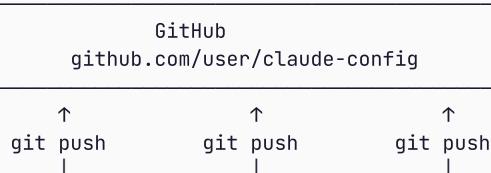
Зачем это нужно?

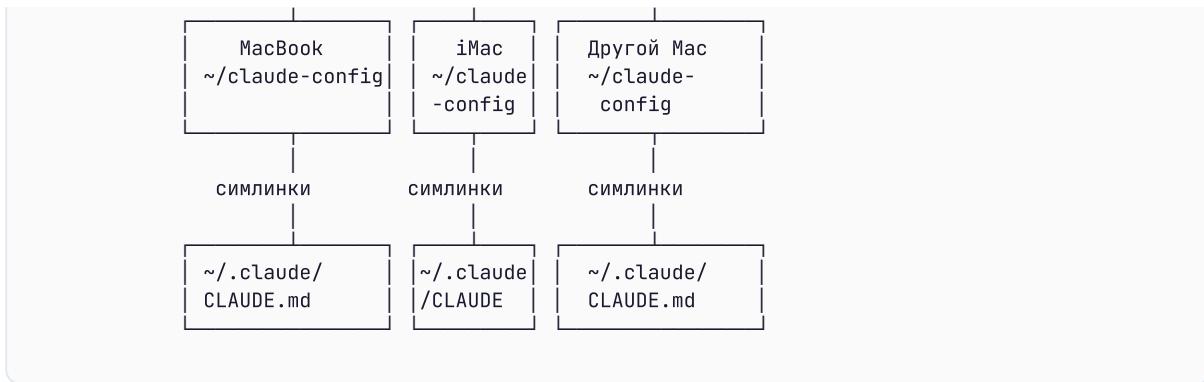
Claude Code ищет настройки в `~/claude/CLAUDE.md`. Но мы хотим хранить их в git-репозитории. Симлинк решает эту проблему:



Как работает синхронизация

Схема потока данных





Принцип работы

1. **Реальные файлы** живут в `~/claude-config/` (git-репозиторий)
2. **Симлинки** в `~/.claude/` указывают на реальные файлы
3. **Claude Code** читает через симлинки, не зная об этом
4. **Git** синхронизирует между компьютерами

Пошаговые сценарии

Сценарий 1: Первая установка на новом компьютере

Ситуация: Купил новый Mac, нужно получить все настройки Claude Code.

```

# 1. Клонируем репозиторий
git clone https://github.com/ungurenko/claude-config.git ~/claude-config

# 2. Задаем права доступа
cd ~/claude-config
chmod +x install.sh

# 3. Устанавливаем
./install.sh

```

Что делает `install.sh`:

- Создаёт папку `~/.claude/` если её нет
- Создаёт симлинки для всех файлов

- Показывает статус установки

Проверка:

```
ls -la ~/.claude/CLAUDE.md  
# Должно показать: CLAUDE.md → /Users/.../claude-config/CLAUDE.md
```

Сценарий 2: Изменил настройки — нужно запушить

Ситуация: Добавил новый скайл или поправил CLAUDE.md.

```
# 1. Переходим в репозиторий  
cd ~/claude-config  
  
# 2. Смотрим что изменилось  
git status  
  
# 3. Добавляем все изменения  
git add .  
  
# 4. Коммитим с описанием  
git commit -m "Add: новый скайл для работы с API"  
  
# 5. Пушим на GitHub  
git push
```

Или одной командой:

```
cd ~/claude-config && git add . && git commit -m "Update: описание" && git
```

Сценарий 3: Получить изменения на другом компьютере

Ситуация: Работал на MacBook, теперь сижу за iMac. Нужны свежие настройки.

```
# 1. Переходим в репозиторий  
cd ~/claude-config
```

```
# 2. Подтягиваем изменения  
git pull  
  
# Готово! Симлинки автоматически видят новые файлы.
```

Важно: Не нужно заново запускать `install.sh` — симлинки уже настроены и указывают на правильные файлы.

Структура репозитория

```
~/claude-config/
├── CLAUDE.md          # Глобальные инструкции
├── agent.md            # Правила поведения
├── install.sh          # Скрипт установки
└── README.md           # Описание репозитория
├── docs/
│   ├── guide.md         # Это руководство
│   └── guide.pdf        # PDF-версия
├── skills/
│   ├── kie-ai/
│   │   └── SKILL.md      # Скилл для Kie.ai
│   └── polza-ai/
│       └── SKILL.md      # Скилл для Polza.ai
└── agents/
    ├── backend-architect.md
    ├── database-optimization.md
    └── debugger.md
```

Памятка команд

Действие	Команда
Первая установка	<code>git clone ... && ./install.sh</code>
Изменил → запушил	<code>git add . && git commit -m "..." && git push</code>
Получить изменения	<code>git pull</code>
Проверить статус	<code>git status</code>
Посмотреть симлинки	<code>ls -la ~/.claude/</code>
История изменений	<code>git log --oneline</code>

FAQ

Что если я изменю файл напрямую в `~/.claude/`?

Это изменит файл в репозитории (через симлинк). Можно спокойно редактировать — git увидит изменения.

Что если симлинк сломался?

Запустите `./install.sh` заново — скрипт пересоздаст симлинки.

Как откатить изменения?

```
cd ~/claude-config  
git checkout -- имя_файла.md
```

Можно ли использовать на Windows?

Да, но симлинки работают иначе. Используйте WSL или Git Bash с правами администратора.

Что если на двух компьютерах разные версии?

Git покажет конфликт при pull. Решите его вручную или:

```
git stash      # Спрятать локальные изменения  
git pull      # Получить новые  
git stash pop # Вернуть спрятанное
```

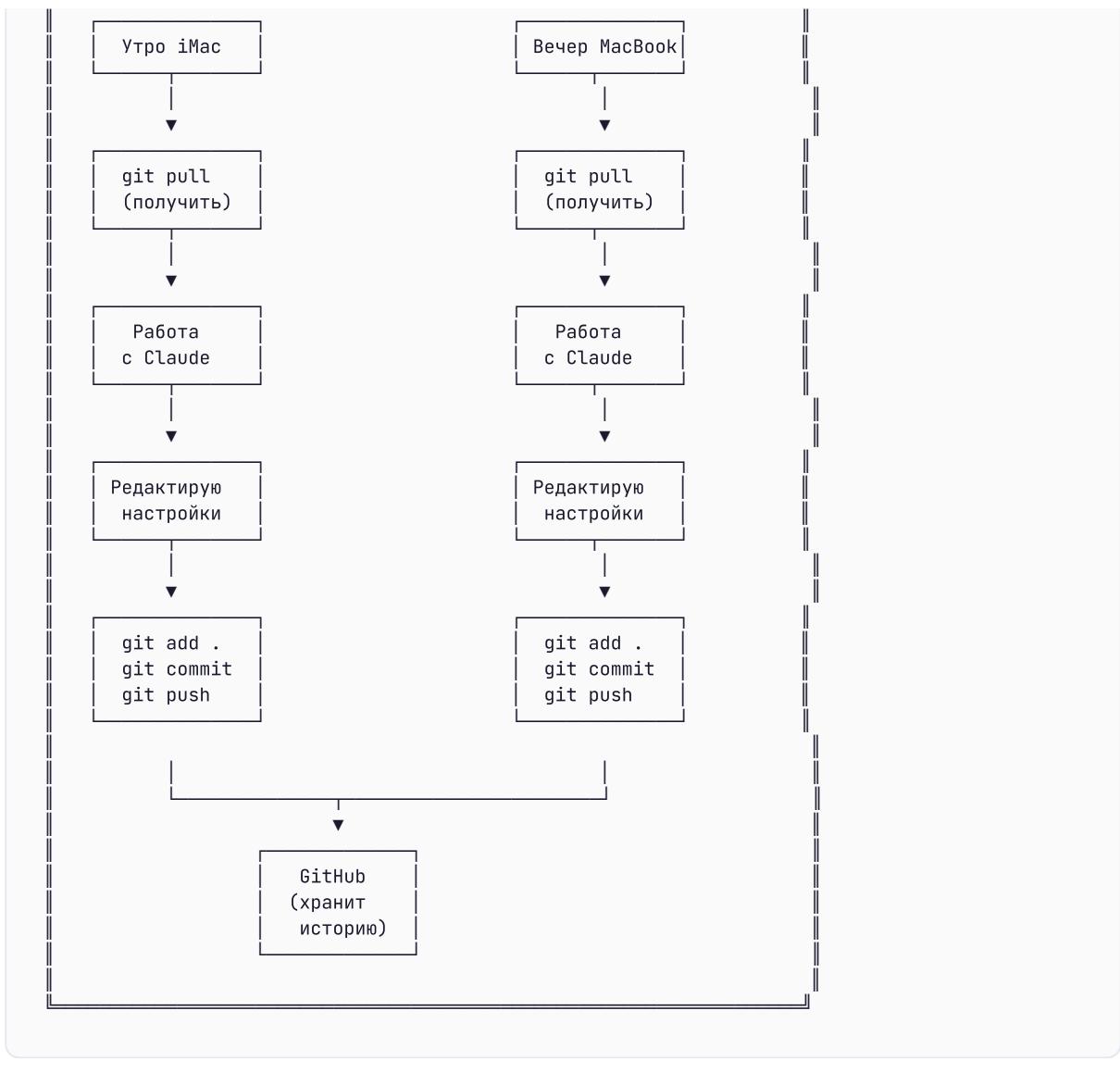
Как добавить новый скайлл?

1. Создайте папку: `mkdir -p skills/my-skill`
2. Создайте файл: `skills/my-skill/SKILL.md`
3. Запустите `./install.sh` (создаст симлинк)
4. Закоммитьте и запушьте

Визуальная схема воркфлоу



ЕЖЕДНЕВНЫЙ ВОРКФЛОУ



Документ создан для репозитория [claude-config](#)