HOOKS

The three hook setup guide

Free guide. No signup. 2 min read

Checked 2026-10-24 against code.claude.com/docs/en/hooks and code.claude.com/docs/en/hooks-guide.

From @learningwithcole

Hooks are small scripts Claude Code runs on its own at set moments. Unlike a line in CLAUDE.md, Claude can't forget or skip them. This guide sets up three:

1. **PreToolUse** runs before a tool call. Exit code 2 blocks the call, and Claude gets your message as feedback.
2. **PostToolUse** runs after a tool call succeeds.
3. **Stop** runs when Claude finishes responding (not when you interrupt it).

You need `jq` for these scripts: `brew install jq` on macOS, `apt-get install jq` on Debian or Ubuntu.

1. The settings file

Put this in `.claude/settings.json` in your project root. That file is for this project only, and you can commit it so your team gets the same rules.

JSON

```
{
  "hooks": {
    "PreToolUse": [
      {
        "matcher": "Edit|Write|Bash",
        "hooks": [
          { "type": "command", "command":
"\"$CLAUDE_PROJECT_DIR\"/.claude/hooks/guard.sh" }
        ]
      }
    ],
    "PostToolUse": [
      {
        "matcher": "Edit|Write",
        "hooks": [
          { "type": "command", "command":
"\"$CLAUDE_PROJECT_DIR\"/.claude/hooks/format.sh" }
        ]
      }
    ],
    "Stop": [
      {
        "hooks": [
          { "type": "command", "command":
"\"$CLAUDE_PROJECT_DIR\"/.claude/hooks/log.sh" }
        ]
      }
    ]
  }
}
```

Already have a `.claude/settings.json`? Add the `hooks` block next to what's there.

2. The three scripts

Save each one in `.claude/hooks/`.

`guard.sh` (PreToolUse): blocks edits to `.env` files and a few dangerous commands.

Terminal

```
#!/bin/bash
INPUT=$(cat)
FILE_PATH=$(echo "$INPUT" | jq -r '.tool_input.file_path // empty')
COMMAND=$(echo "$INPUT" | jq -r '.tool_input.command // empty')

if [[ "$FILE_PATH" == *".env"* ]]; then
  echo "Blocked: $FILE_PATH is a protected .env file" >&2
  exit 2
fi

for pattern in "rm -rf" "git push --force" "git reset --hard"; do
  if [[ "$COMMAND" == *"$pattern"* ]]; then
    echo "Blocked: '$pattern' is not allowed in this project" >&2
    exit 2
  fi
done

exit 0
```

`format.sh` (PostToolUse): runs Prettier on the file Claude just edited. Swap in your own formatter (`black`, `gofmt`, `rustfmt`) if you don't use Prettier.

Terminal

```
#!/bin/bash
FILE_PATH=$(jq -r '.tool_input.file_path // empty')
[ -n "$FILE_PATH" ] && npx prettier --write "$FILE_PATH" >/dev/null 2>&1
exit 0
```

`log.sh` (Stop): adds a short entry to `.claude/session.log` each time Claude finishes: the time, a one-line git diff summary, and the start of Claude's last reply.

Terminal

```
#!/bin/bash
INPUT=$(cat)
# Skip if Claude is already continuing because of a Stop hook
[ "$(echo "$INPUT" | jq -r '.stop_hook_active')" = "true" ] && exit 0

{
  echo "## $(date '+%Y-%m-%d %H:%M')"
  git -C "$CLAUDE_PROJECT_DIR" diff --stat 2>/dev/null | tail -n 1
  echo "$INPUT" | jq -r '.last_assistant_message // empty' | head -c 400
  echo
  echo
} >> "$CLAUDE_PROJECT_DIR/.claude/session.log"

exit 0
```

Add `.claude/session.log` to your `.gitignore` if you don't want it in the repo.

Then make all three executable (hooks won't run otherwise):

Terminal

```
chmod +x .claude/hooks/guard.sh .claude/hooks/format.sh .claude/hooks/log.sh
```

3. Test it

1. Start Claude Code in the project and type `/hooks`. You should see one hook each under PreToolUse, PostToolUse and Stop.
2. Ask Claude: "add a comment to my .env file". The edit is blocked before it runs, and Claude sees "Blocked: … is a protected .env file".
3. Ask Claude to change a line in any JS or TS file, then open it. Prettier has reformatted it.
4. Open `.claude/session.log`. There's an entry for each reply.

Good to know

- A hook can't loosen your permission rules, and a block holds in every permission mode.
- The Bash check is a pattern match, so it's a guard rail, not a wall. For a hard "never run this", also add a deny rule in your permissions.

- To turn all hooks off for a while without deleting them, add `"disableAllHooks": true` to the settings file.