April 13, 2026 · 7 min read · By Yiğit Konur

Claude Code plan mode'u PermissionRequest ile auto-approve etmek

ai software-engineering

Claude Code'un plan mode'unu kullanıyorsan o "ready to code?" onay dialog'unu büyük ihtimalle görmüşsündür. ben Opus 4.6'nın planını zaten her seferinde onaylıyorum, yani artık bu işi elle yapmaya gerek yok. doğru event'i hedefleyince o tıklamayı tek bir hook ile otomatikleştirebiliyorsun.

bu aynı zamanda anthropic'in repo'sunda takip edilen küçük bir docs gap'ini de görünür kılıyor: issue #11891 (https://github.com/anthropics/claude-code/issues/11891) ("[DOCS] Missing PermissionRequest hook details in Hooks Guide and Input Reference schema"). + iyi documente edilmemiş ve JSON payload detayları konusunda daha iyi bir iş çıkarmamız lazım; Claude Code open-source olmadığı için temel bir reverse engineering yapmak zorunda kaldık (ismi kadar havalı değil, temel olarak sadece JSON'ları bir klasöre kaydeden bir event listener ekleyip oradan okuyorsun, assembly patch'lenmiş değil xD)

asıl detay: Stop değil, PermissionRequest

onay dialog'u, `ExitPlanMode` tool'u için gelen bir `PermissionRequest`.

`Stop` ise run bittikten sonra tetikleniyor. plan onayı Claude hala kullanıcıdan input *beklerken* oluyor, yani `Stop` burayı yakalamak için yanlış yer.

PermissionRequest stdin'e ne gönderiyor

dialog açıldığında stdin'den gelen şekil (şu anda tam documente değil) şu:

-

```json
{
  "session_id": "abc123-def4-5678-ghij-klmnopqrstuv",
  "transcript_path": "/Users/you/.claude/projects/-Users-you-my-project/abc123.jsonl",
  "cwd": "/Users/you/my-project",
  "permission_mode": "plan",
  "hook_event_name": "PermissionRequest",
  "tool_name": "ExitPlanMode",
  "tool_input": {
    "allowedPrompts": [
      { "tool": "Bash", "prompt": "run tests" },
      { "tool": "Bash", "prompt": "install deps" }
    ],
    "plan": "# Plan: Implement Feature X\n\n## Context\n\n..."
  }
}
```

işine yarayacak iki parça:

- `tool_name` sana *ne* sorduğunu söylüyor (burada: `ExitPlanMode`)
- `tool_input.plan` plan markdown'ının tamamını taşıyor (başka bir yere arşivleyebilirsin)

minimum auto-approve hook (kopyala/yapıştır)

script: `~/.claude/hooks/auto-approve-plan.sh`

```bash
#!/bin/bash
cat >/dev/null  # consume stdin (important on Windows/WSL)
cat <<'EOF'
{"hookSpecificOutput":{"hookEventName":"PermissionRequest","decision":{"behavior":"all
EOF
```

config (her şeyi değil, sadece `ExitPlanMode`'u match et):

-

json

```
{
  "hooks": {
    "PermissionRequest": [
      {
        "matcher": "ExitPlanMode",
        "hooks": [
          {
            "type": "command",
            "command": "~/.claude/hooks/auto-approve-plan.sh"
          }
        ]
      }
    ]
  }
}
```

vakit kazandıracak notlar:

- hook'un stdin'i okusun (yoksa takılabilirsin)
- output'un `hookSpecificOutput` altında, `hookEventName: "PermissionRequest"` ile sarılı olmalı

opsiyonel: her planı Craft.do'ya arşivle

plan markdown'ı zaten `tool_input.plan` içinde olduğu için onu arka planda Craft.do'ya atıp onay cevabını anında döndürebilirsin.

tek "gotcha" şu: markdown'ı bir shell değişkenine çekip sonra JSON'a tekrar gömme (multi-line içerik seni vuracaktır). Craft.do payload'ını tek bir `jq` pass'iyle kur, olsun bitsin.

bash

```
#!/bin/bash
TMPFILE="$(mktemp)"
cat > "$TMPFILE"
```

-

fire-and-forget arşiv (onay cevabını blokeleme)

```bash
(
jq
--arg ts "$(date '+%Y-%m-%d %H:%M')"
--arg pageId "$CRAFT_PAGE_ID"
--arg home "$HOME"
'{
blocks: [{
type: "page",
textStyle: "card",
markdown: ("[" + (.cwd | sub($home; "~")) + "] - [" + $ts + "]"),
content: [{ type: "text", markdown: .tool_input.plan }]
}],
position: { position: "end", pageId: $pageId }
}' < "$TMPFILE"
| curl -sS -X POST "$CRAFT_API_URL/blocks" -H "Content-Type: application/json" -d @-
>/dev/null 2>&1
) &

rm -f "$TMPFILE"
```

dialog'u onayla

```bash
cat <<'EOF'
{"hookSpecificOutput":{"hookEventName":"PermissionRequest","decision":{"behavior":"all
EOF
```

paketli hali istersen: yigitkonur/hooks-claude-approve

"auto-approve + (opsiyonel) Craft.do arşivi" akışını yigitkonur/hooks-claude-approve (https://github.com/yigitkonur/hooks-claude-approve) içine topladım.

-

mode'lar:

kurulum:

```bash
bash <(curl -fsSL https://raw.githubusercontent.com/yigitkonur/hooks-claude-approve/ma
```

link'ler

- hooks tool: https://github.com/yigitkonur/hooks-claude-approve (https://github.com/yigitkonur/hooks-claude-approve)
- docs gap tracker: https://github.com/anthropics/claude-code/issues/11891 (https://github.com/anthropics/claude-code/issues/11891)
- official hooks guide: https://docs.anthropic.com/en/docs/claude-code/hooks (https://docs.anthropic.com/en/docs/claude-code/hooks)

SOURCES

Anthropic hooks guide (https://docs.anthropic.com/en/docs/claude-code/hooks)

Docs gap issue #11891 (https://github.com/anthropics/claude-code/issues/11891)

hooks-claude-approve (https://github.com/yigitkonur/hooks-claude-approve)

-