

## Setup Claude Code with Group available LLMs

Identify the current list of Available Models

Setup Guide

Installation

Authentication

Configuration

Model Name

Token Allocation Strategy

Customisation

Model Compatibility

Testing Methodology

Latest Test Results

Model Common Error Messages


Limitations


Models

Web Search

Claude Common Errors


### Identify the current list of Available Models

 As with everything in the Group, CLI connectivity depends on the correct proxy and SSL settings.

The easy way → You can head there and let this script sort it for you on MacOS :  [Pro](#)

[xy and SSL Certificate issues](#)

▼ or the manual way...

 As with everything in the Group, CLI connectivity depends on the correct proxy and SSL settings.

- To solve proxy issues, run Alpaca:

 [Setting up Alpaca on MacOS](#)


 [Setting up Alpaca on Windows](#)

```
2025/09/22 15:36:00.993448 proxyfinder.go:135: [82] CONNECT //api.anthropic.com:443 via "PROXY cba.proxy.prismaaccess.com:8080"
2025/09/22 15:36:08.538248 proxyfinder.go:113: [83] CONNECT //api.studio.genai.cba:443 via "DIRECT"
2025/09/22 15:36:08.712029 proxyfinder.go:135: [84] CONNECT //statsig.anthropic.com:443 via "PROXY cba.proxy.prismaaccess.com:8080"
2025/09/22 15:36:16.273745 proxyfinder.go:113: [85] CONNECT //api.studio.genai.cba:443 via "DIRECT"
2025/09/22 15:36:16.488873 proxyfinder.go:135: [86] CONNECT //statsig.anthropic.com:443 via "PROXY cba.proxy.prismaaccess.com:8080"
2025/09/22 15:36:31.004767 proxyfinder.go:135: [87] CONNECT //api.anthropic.com:443 via "PROXY cba.proxy.prismaaccess.com:8080"
2025/09/22 15:36:46.364294 proxyfinder.go:113: [88] CONNECT //api.studio.genai.cba:443 via "DIRECT"
2025/09/22 15:36:46.382537 proxyfinder.go:113: [89] CONNECT //api.studio.genai.cba:443 via "DIRECT"
2025/09/22 15:36:49.554551 proxyfinder.go:135: [90] CONNECT //registry.npmjs.org:443 via "PROXY cba.proxy.prismaaccess.com:8080"
2025/09/22 15:37:04.161193 proxyfinder.go:113: [91] CONNECT //api.studio.genai.cba:443 via "DIRECT"
2025/09/22 15:37:22.346531 proxyfinder.go:113: [92] CONNECT //api.studio.genai.cba:443 via "DIRECT"
2025/09/22 15:37:56.436576 proxyfinder.go:113: [93] CONNECT //api.studio.genai.cba:443 via "DIRECT"
```

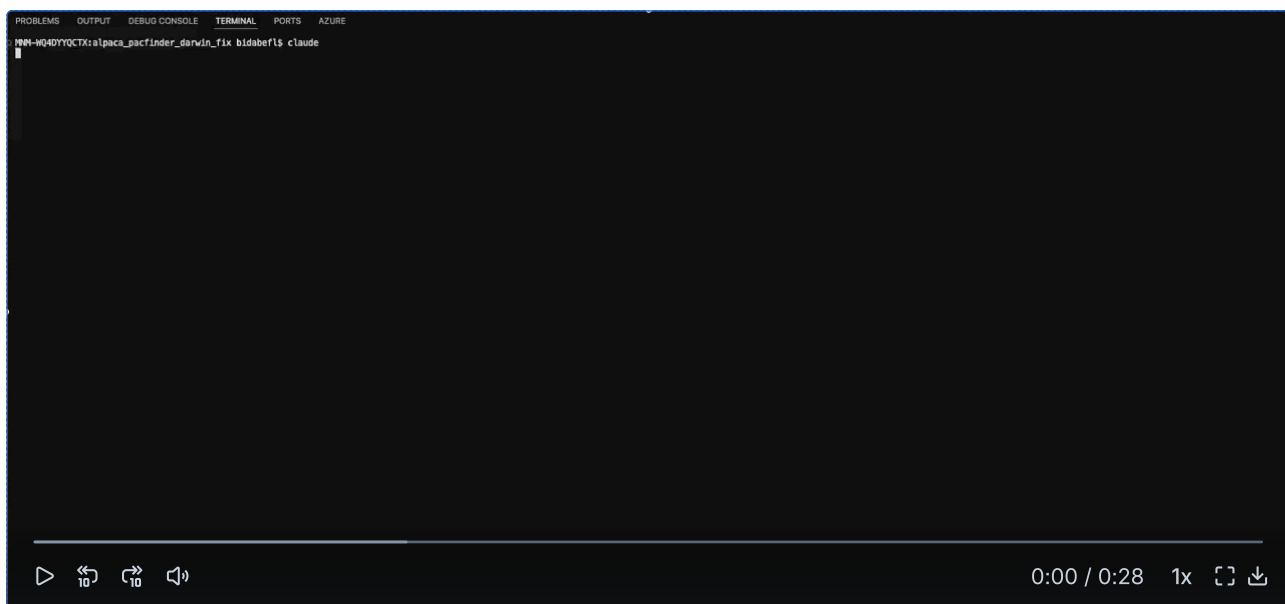
- This is because Claude Code requires traffic to go sometimes direct **and** sometimes via Prisma... Claude was ignoring the [no\\_proxy variable](#), so Alpaca is the way to apply the right web routing logic!

- To solve SSL issues, you should build a PEM certificate store and point your CLI to it using environment variables... Notably `NODE_EXTRA_CA_CERTS` needs to be declared

 [Group Internal Root CAs Certificate Store](#)

 Request `AR - GenAI Studio - Prod - User` access in [identity.cba](#) to access it at: <https://studio.genai.cba> and get an API Key

 The below was tested on Claude Code `1.0.40`, LiteLLM `1.72.2` ([api.studio.genai.cba](#)), MacOS `15.5 (24F74)`, zsh `5.9` & bash `3.2.57(1)`



## Setup Guide

### Installation

Follow the official documentation for the most up-to-date installation instructions: [Set up Claude Code - Claude Docs](#)

Current installation command: `npm install -g @anthropic-ai/claude-code`

Alternative: `brew install claude` (requires admin privilege as it creates a Claude.app in `/Applications`)

```
~/Github | main !10 713 | which claude
/opt/homebrew/bin/claude
```

Either way, whether using `npm` or `brew`, the binary comes from Homebrew

## Authentication

Follow the official documentation for the most up-to-date authentication instructions: [🔥 LLM gateway configuration - Claude Docs](#)

1. Update/Save the `sk-` key in the Keychain:

```
security delete-generic-password -s "LITELLM_API_KEY";
security add-generic-password -s "LITELLM_API_KEY" -a
"$(whoami)" -w "KEY"
```

where KEY is your `sk-` key from Open-WebUI (<https://studio.genai.cba>)

2. Add the following line to your shell configuration file (`~/.zshrc`, `~/.bashrc`, etc.)

```
1 export LITELLM_MASTER_KEY=$(security find-generic-password -s "LITELLM_API_KEY" -w)
2 export ANTHROPIC_AUTH_TOKEN="$LITELLM_MASTER_KEY"
```

3. Restart your terminal or run `source ~/.zshrc` (or equivalent for your shell)

## Configuration

Create or update your `~/.claude/settings.json` file with the following configuration:

**Edit: updated on** 28 Aug 2025

```
1 {
2   "permissions": {
3     "allow": [
4       "Bash(npm run lint)",
5       "Bash(npm run test:*)",
6       "Bash(npm run build)",
7       "Bash(npm run dev)",
8       "Bash(git status)",
```

```

9      "Bash(git diff)",
10     "Bash(git log --oneline -10)",
11     "Read(~/.zshrc)",
12     "Read(~/.gitconfig)",
13     "Read(package.json)",
14     "Read(tsconfig.json)",
15     "Read(.eslintrc*)",
16     "Read(.prettierrc*)",
17     "Read(*.md)"
18 ],
19 "deny": [
20     "Bash(curl:*)",
21     "Bash(wget:*)",
22     "Bash(rm -rf:*)",
23     "Bash(sudo:*)",
24     "Write(~/.ssh/*)",
25     "Write(/etc/*)"
26 ]
27 },
28 "env": {
29     "CLAUDE_CODE_ENABLE_TELEMETRY": "0",
30     "ANTHROPIC_BASE_URL": "https://api.studio.genai.cba",
31     "ANTHROPIC_MODEL": "aipe-bedrock-claude-4-sonnet",
32     "CLAUDE_CODE_MAX_OUTPUT_TOKENS": "32000",
33     "MAX_THINKING_TOKENS": "16000",
34     "DISABLE_PROMPT_CACHING": "0",
35     "CLAUDE_CODE_TEMPERATURE": "0.1",
36     "CLAUDE_CODE_MAX_INPUT_TOKENS": "150000"
37 }
38 }

```

#### Model Name

Replace `[model-name]` with one of the compatible models listed below:

`"ANTHROPIC_MODEL": "bedrock-claude-3-5-sonnet-v2"`, or `"aipe-bedrock-claude-4-sonnet"`

See the full list of supported models below: [🔗 Setup Claude Code with Group available LLMs | L](#)  
[atest Test Results \(Updated \[date\] \)](#)



- Set `DISABLE_PROMPT_CACHING` to "1" or you'll get this:

```

~/Github/ClaudeCode | main 16 711 | claudes -p "What's your model?" --system-prompt "You are a senior backend engineer"
API Error: 400 {"error":{"message":"{\\\"message\\\":\\\"You invoked an unsupported model or your request did not allow prompt caching. See the documentation for more information.\\\"}\\\",\\\"type\\\":\\\"None\\\",\\\"param\\\":\\\"None\\\",\\\"code\\\":\\\"400\\\"}}
~/Github/ClaudeCode | main 16 711 |

```

#### Token Allocation Strategy

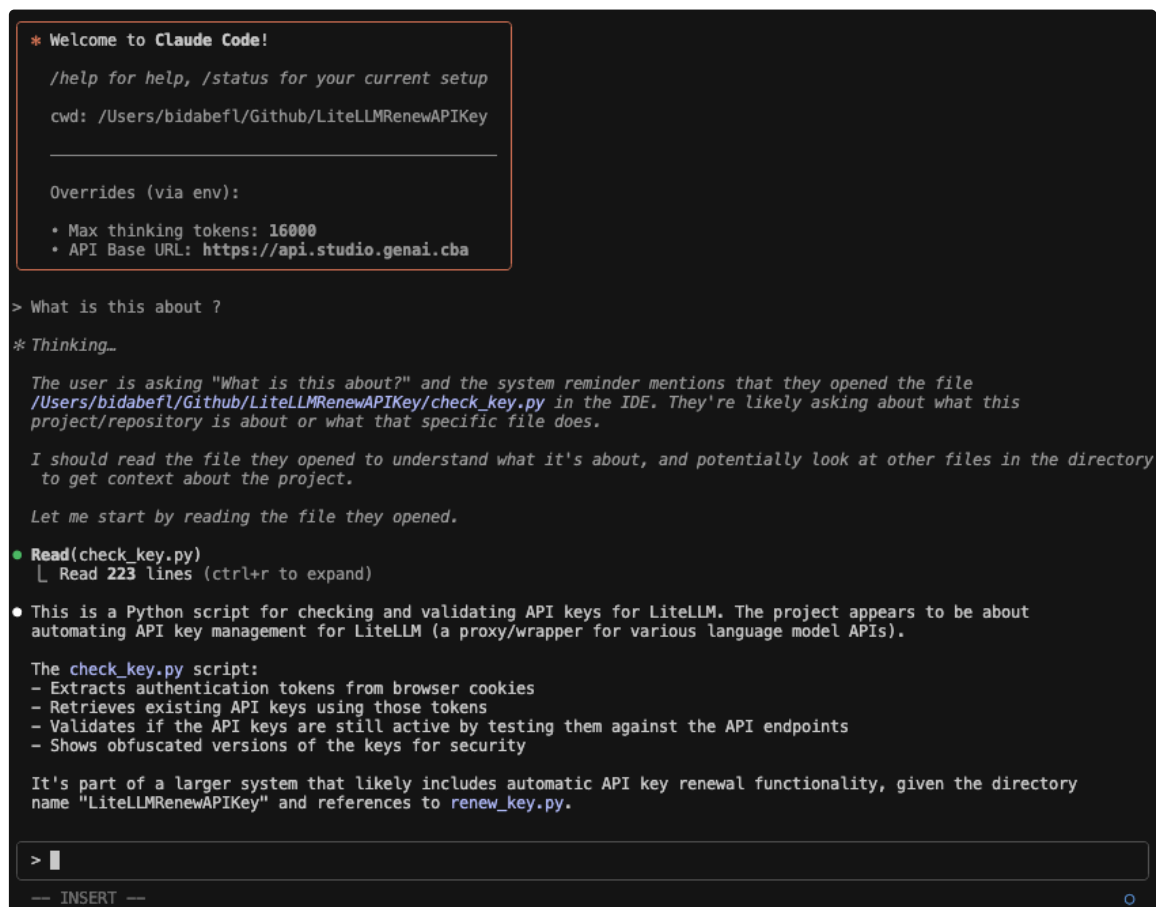
These token limits optimise performance and cost whilst maximising the model's capabilities.

`MAX_THINKING_TOKENS` : "16000" allows the model substantial reasoning capacity for complex problem-solving,

whilst `CLAUDE_CODE_MAX_OUTPUT_TOKENS : "32000"` provides ample space for comprehensive responses, code generation, and detailed explanations.

Together, they utilise 48K of the available 64K output budget, leaving a safe buffer.

`CLAUDE_CODE_MAX_INPUT_TOKENS : "150000"` maintains 75% of the maximum input capacity (200K), providing extensive context for large codebases whilst controlling costs—saving approximately 25% on input token expenses compared to using the full limit, yet still allowing roughly 300-400 pages worth of context per request.



```
* Welcome to Claude Code!

/help for help, /status for your current setup
cwd: /Users/bidabefl/Github/LiteLLMRenewAPIKey

-----

Overrides (via env):
• Max thinking tokens: 16000
• API Base URL: https://api.studio.genai.cba

> What is this about ?

* Thinking...

The user is asking "What is this about?" and the system reminder mentions that they opened the file
/Users/bidabefl/Github/LiteLLMRenewAPIKey/check_key.py in the IDE. They're likely asking about what this
project/repository is about or what that specific file does.

I should read the file they opened to understand what it's about, and potentially look at other files in the directory
to get context about the project.

Let me start by reading the file they opened.

• Read(check_key.py)
  └─ Read 223 lines (ctrl+r to expand)

• This is a Python script for checking and validating API keys for LiteLLM. The project appears to be about
  automating API key management for LiteLLM (a proxy/wrapper for various language model APIs).

  The check_key.py script:
  - Extracts authentication tokens from browser cookies
  - Retrieves existing API keys using those tokens
  - Validates if the API keys are still active by testing them against the API endpoints
  - Shows obfuscated versions of the keys for security

  It's part of a larger system that likely includes automatic API key renewal functionality, given the directory
  name "LiteLLMRenewAPIKey" and references to renew_key.py.

> |
-- INSERT --
```

These settings provide high quality LLM responses!

Where did I get these figures from? See the hidens section below...

✓ Expand for more information...

```
1 curl -X 'GET' \
2   'https://api.studio.genai.cba/api/v1/v1/model/info?
   litellm_model_id=8811c446ab141d8b386f6b25005f4cc9ced7f3a5268371f07a7dcd4f50e06590' \
3   -H 'accept: application/json' \
4   -H 'x-litellm-api-key: sk-your_key...not_mine'
```

returns

```

1  {
2    "data": [
3      {
4        "model_name": "aipe-bedrock-claude-4-sonnet",
5        "litellm_params": {
6          "aws_region_name": "us-east-2",
7          "use_in_pass_through": false,
8          "use_litellm_proxy": false,
9          "merge_reasoning_content_in_choices": false,
10         "model": "bedrock/converse/us.anthropic.claude-sonnet-4-20250514-v1:0",
11         "aws_role_name": "arn:aws:iam::910696201088:role/studio-target-prd-bedrock-role",
12         "aws_session_name": "cross-dhp-cto-role",
13         "model_id": "arn:aws:bedrock:us-east-2:910696201088:inference-
profile/us.anthropic.claude-sonnet-4-20250514-v1:0"
14       },
15       "model_info": {
16         "id": "8811c446ab141d8b386f6b25005f4cc9ced7f3a5268371f07a7dcd4f50e06590",
17         "db_model": false,
18         "key": "anthropic.claude-sonnet-4-20250514-v1:0",
19         "max_tokens": 64000,
20         "max_input_tokens": 200000,
21         "max_output_tokens": 64000,
22         "input_cost_per_token": 0.000003,
23         "cache_creation_input_token_cost": 0.00000375,
24         "cache_read_input_token_cost": 3e-7,
25         "input_cost_per_character": null,
26         "input_cost_per_token_above_128k_tokens": null,
27         "input_cost_per_token_above_200k_tokens": null,
28         "input_cost_per_query": null,
29         "input_cost_per_second": null,
30         "input_cost_per_audio_token": null,
31         "input_cost_per_token_batches": null,
32         "output_cost_per_token_batches": null,
33         "output_cost_per_token": 0.000015,
34         "output_cost_per_audio_token": null,
35         "output_cost_per_character": null,
36         "output_cost_per_reasoning_token": null,
37         "output_cost_per_token_above_128k_tokens": null,
38         "output_cost_per_character_above_128k_tokens": null,
39         "output_cost_per_token_above_200k_tokens": null,
40         "output_cost_per_second": null,
41         "output_cost_per_image": null,
42         "output_vector_size": null,
43         "litellm_provider": "bedrock_converse",
44         "mode": "chat",
45         "supports_system_messages": null,
46         "supports_response_schema": true,
47         "supports_vision": true,
48         "supports_function_calling": true,
49         "supports_tool_choice": true,
50         "supports_assistant_prefill": true,
51         "supports_prompt_caching": true,
52         "supports_audio_input": null,
53         "supports_audio_output": null,
54         "supports_pdf_input": true,
55         "supports_embedding_image_input": null,
56         "supports_native_streaming": null,
57         "supports_web_search": null,

```

```

58     "supports_url_context": null,
59     "supports_reasoning": true,
60     "supports_computer_use": true,
61     "search_context_cost_per_query": {
62         "search_context_size_low": 0.01,
63         "search_context_size_medium": 0.01,
64         "search_context_size_high": 0.01
65     },
66     "tpm": null,
67     "rpm": null,
68     "supported_openai_params": [
69         "max_tokens",
70         "max_completion_tokens",
71         "stream",
72         "stream_options",
73         "stop",
74         "temperature",
75         "top_p",
76         "extra_headers",
77         "response_format",
78         "tools",
79         "thinking",
80         "reasoning_effort"
81     ]
82 }
83 }
84 ]
85 }

```

## Customisation

✧ [Optimize your terminal setup - Claude Docs](#)

✧ [Add Claude Code to your IDE - Claude Docs](#)

## Model Compatibility


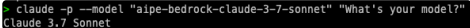
### Testing Methodology

Each model was tested using the following procedure:

1. Setting the model name in `~/.claude/settings.json`
2. Running `claude -p "What's your model?"` to test if the model responds correctly
3. Recording the response or error message

## Latest Test Results

*Edit: Updated on* 14 Jul 2025

Model	Status	Error
bedrock-claude-3-5-sonnet-v2	✓	Working
aip-bedrock-claude-3-5-sonnet-v2	✓	Working
bedrock-claude-3-7-sonnet	✗	Cannot read properties of undefined (reading 'filter')
bedrock-claude-4-sonnet	✗	Cannot read properties of undefined (reading 'filter')
aip-bedrock-claude-4-sonnet	✓	<del>Cannot read properties of undefined (reading 'filter')</del> Sweet... It now works! 
aip-bedrock-claude-3-7-sonnet	✓	<del>Cannot read properties of undefined (reading 'filter')</del> Ditto, now works! 
bedrock-claude-3-sonnet	✗	system: Input should be a valid string
guardrails-bedrock-claude-3-5-sonnet-v2	✗	LLM Provider NOT provided

According to [the official documentation](#), the following model providers are supported through LiteLLM gateway:

- Anthropic (Claude models)
- Amazon Bedrock
- Google Vertex AI

However, our testing shows that only specific models within these providers are compatible.

Most models found in [Available Models in the Group](#) are not supported in Claude Code (to-date)



The other Bedrock models above should be supported, but I suspect this might solved itself with updated versions of our LiteLLM instance ([api.studio.genai.cba](https://api.studio.genai.cba), currently version 1.72.2)

**Model Common Error Messages**

When using unsupported models, you may encounter these error messages:

Error Type	Message	Explanation
<b>Filter Error</b>	Cannot read properties of undefined (reading 'filter')	The model is not properly configured in the gateway
<b>LLM Provider Error</b>	Error calling litellm.acompletion for non-Anthropic model: LLM Provider NOT provided	The gateway doesn't recognize the model provider
<b>Input Validation Error</b>	system: Input should be a valid string	The model cannot process the input format
<b>Invalid Model Name</b>	Invalid model name passed in model=[model name]	The model name is not recognized
<b>Unsupported Model</b>	You invoked an unsupported model or your request did not allow prompt caching	The model is not supported or prompt caching is enabled

**Limitations**

**Models**

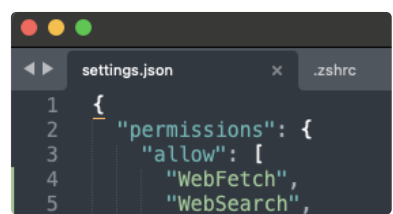
It appears Claude only supports Anthropics models (Claude), unlike Codex and Gemini which supports a lot more...

[Setup Gemini CLI with Group available LLMs](#)

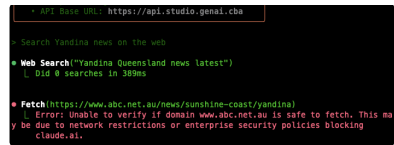
 [Setup Codex with GPT-5](#)

Web Search

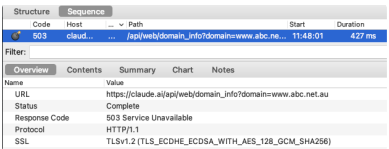
Whilst Claude Code supports web search




Web Search Settings



Attempted Web Search



Claude Code passes the Web Traffic to [claude.ai](https://claude.ai), which our PaloAlto network appliance blocks...

Unlike Codex, it does not send a cURL request then parse its response. So Web Search is 

Claude Common Errors

Error Type	Message	Explanation
------------	---------	-------------



(Invalid settings file in reality)

```
Received API Key =
sk-...h_gw, Key Hash
(Token) =. Unable to find
token in cache or
LiteLLM_VerificationToken
able", "type": "token_not_fo
und_in_db", "param": "key", "
code": "401"}}) · Retrying
in 5 seconds...
```

```
Welcome to Claude Code
/help for help, /status for your current setup
cwd: /Users/9iddaef/Github/litellm-key-updater

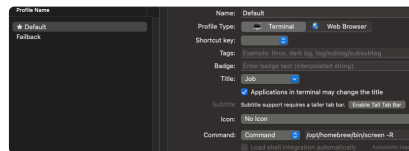
Overrides (via env):
• Max thinking tokens: 15000
• API Base URL: https://api.studio.gemini.cha

Hello
• API Error (401) {"error":{"message":"Authentication Error, Invalid proxy server token passed. Received API Key = sk-...h_gw, Key H
sh (Token) =2883a115d261d25e596800319d370ba27f5ca8b61b0d8c8a237d6271ae8, Unable to find token in cache or 'LiteLLM_Verification
Token'","type":"token_not_found_in_db","param":"key","code":"401"}} Retrying in 5 seconds (attempt 4/10)
```

## Display / UI

No Message, but Claude  
stdout and stderr is hardly  
readable...

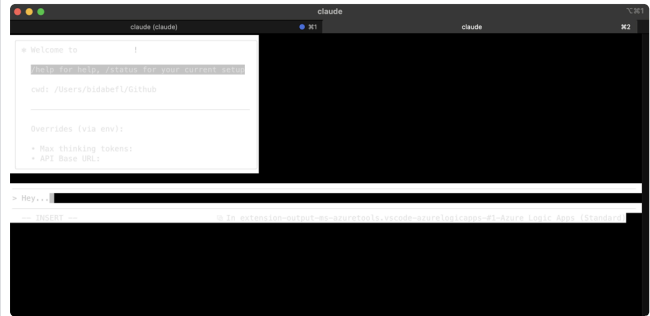
The issue was caused by  
[GNU Screen](#)



(if you're using it) as I am...

Changing the terminal color,  
themes and other visual  
effects did not solve this  
issue.

Updating `screen` using  
`brew upgrade`  
`screen` fixed the issue.  
Alternatively, do not call  
`claude` within `screen`  
(e.g. other profile)



🎉 Helpful? Drop me a thanks on [Achievers](#) ! And if you've got knowledge to share, don't hold back - we all grow when we learn from each other 💡