

Agent/Client Protocol

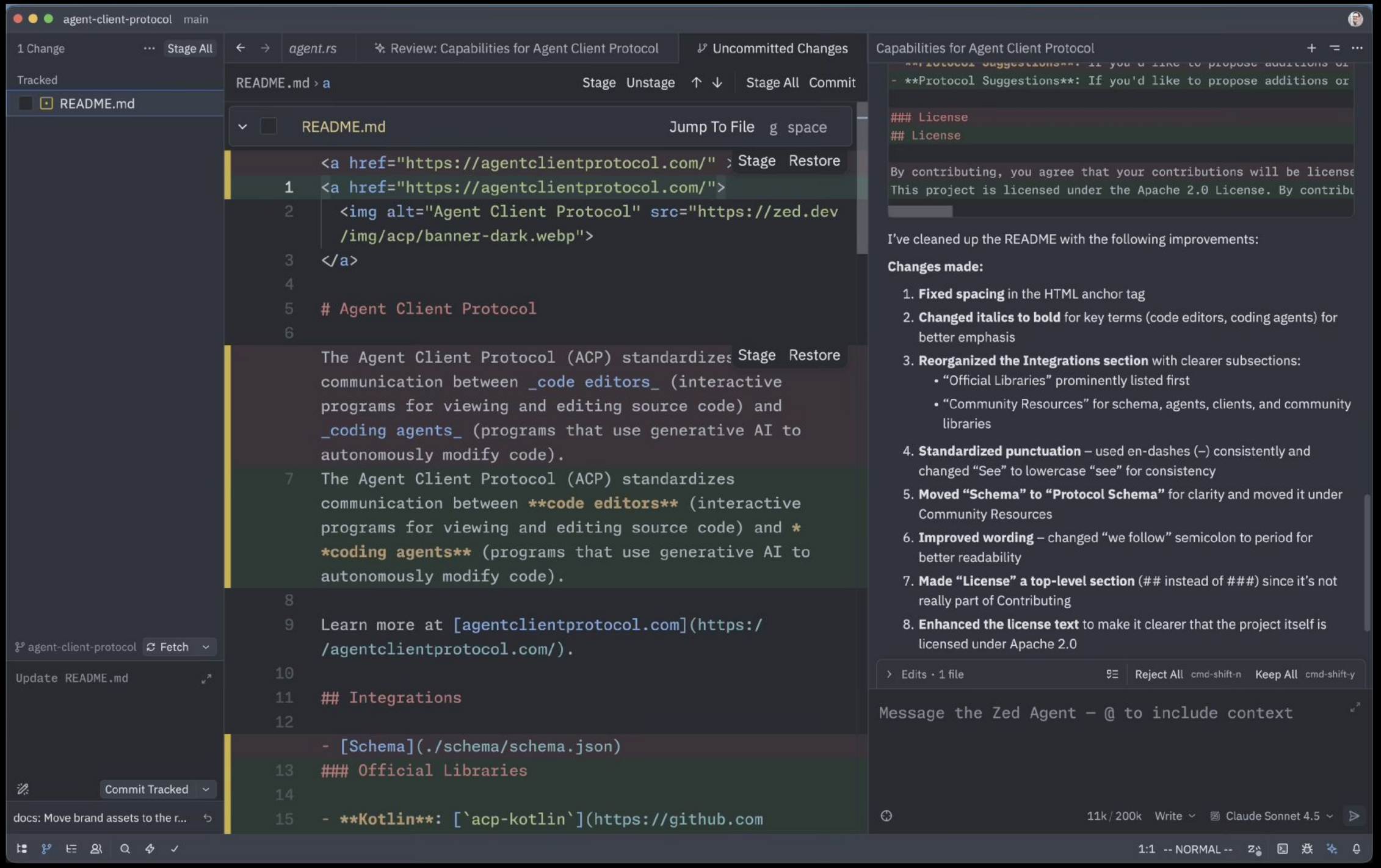
Ben Brandt, Zed
Sergey Ignatov, JetBrains



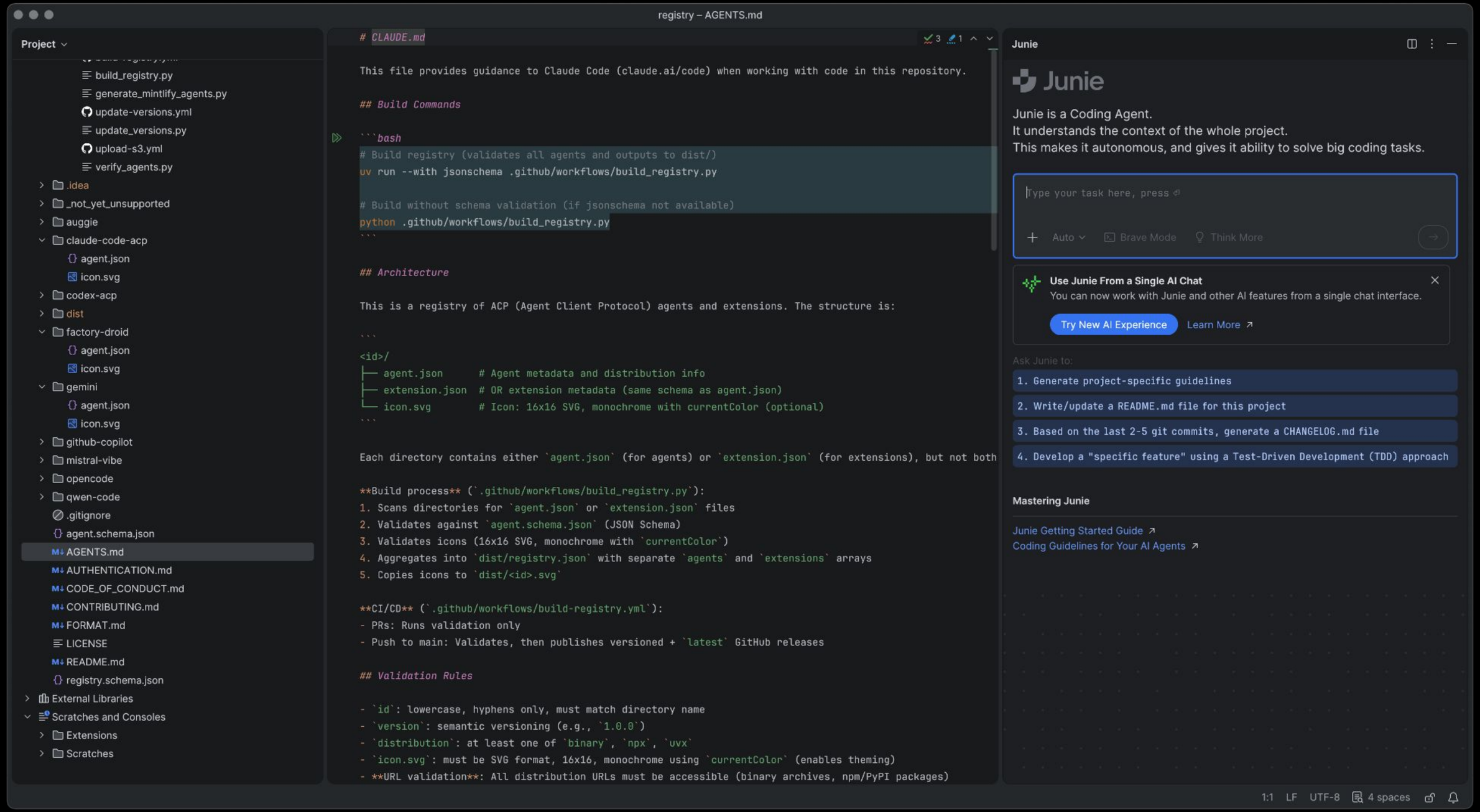
JETBRAINS

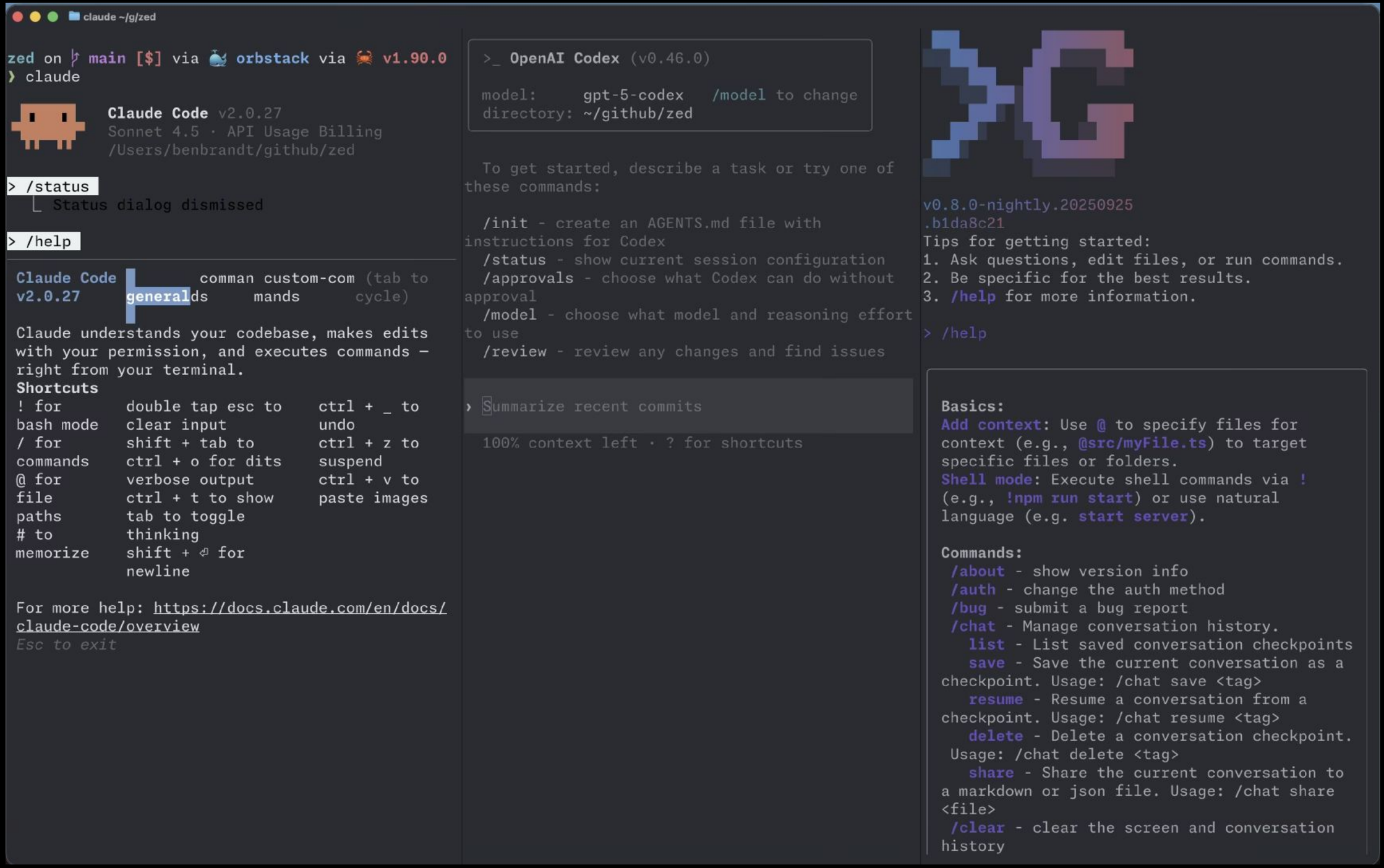


Zed


agent-client-protocol main
1 Change
Stage All
Tracked
README.md
agent.rs
Review: Capabilities for Agent Client Protocol
Uncommitted Changes
README.md › a
Stage Unstage
Stage All Commit
README.md
Jump To File g space

Stage Restore
1
2 <img alt="Agent Client Protocol" src="https://zed.dev
/img/acp/banner-dark.webp">
3
4
5 # Agent Client Protocol
6
The Agent Client Protocol (ACP) standardizes
Stage Restore
communication between _code editors_ (interactive
programs for viewing and editing source code) and
coding agents (programs that use generative AI to
autonomously modify code).
7 The Agent Client Protocol (ACP) standardizes
communication between **code editors** (interactive
programs for viewing and editing source code) and *
*coding agents** (programs that use generative AI to
autonomously modify code).
8
9 Learn more at [agentclientprotocol.com](https:/
/agentclientprotocol.com/).
10
11 ## Integrations
12
- [Schema](./schema/schema.json)
13 ### Official Libraries
14
15 - **Kotlin**: [`acp-kotlin`](https://github.com
agent-client-protocol
Fetch
Update README.md
Commit Tracked
docs: Move brand assets to the r...
Capabilities for Agent Client Protocol
- **Protocol Suggestions**: If you'd like to propose additions or
License
License
By contributing, you agree that your contributions will be license
This project is licensed under the Apache 2.0 License. By contribu
I've cleaned up the README with the following improvements:
Changes made:
1. Fixed spacing in the HTML anchor tag
2. Changed italics to bold for key terms (code editors, coding agents) for better emphasis
3. Reorganized the Integrations section with clearer subsections:
• "Official Libraries" prominently listed first
• "Community Resources" for schema, agents, clients, and community libraries
4. Standardized punctuation – used en-dashes (–) consistently and changed "See" to lowercase "see" for consistency
5. Moved "Schema" to "Protocol Schema" for clarity and moved it under Community Resources
6. Improved wording – changed "we follow" semicolon to period for better readability
7. Made "License" a top-level section (## instead of ###) since it's not really part of Contributing
8. Enhanced the license text to make it clearer that the project itself is licensed under Apache 2.0
Edits · 1 file
Reject All cmd-shift-n
Keep All cmd-shift-y
Message the Zed Agent – @ to include context
11k / 200k
Write
Claude Sonnet 4.5
1:1 -- NORMAL --


registry – AGENTS.md
Project
build_registry.py
generate_mintlify_agents.py
update-versions.yml
update_versions.py
upload-s3.yml
verify_agents.py
.idea
_not_yet_unsupported
auggie
claude-code-acp
agent.json
icon.svg
codex-acp
dist
factory-droid
agent.json
icon.svg
gemini
agent.json
icon.svg
github-copilot
mistral-vibe
opencode
qwen-code
.gitignore
agent.schema.json
AGENTS.md
AUTHENTICATION.md
CODE_OF_CONDUCT.md
CONTRIBUTING.md
FORMAT.md
LICENSE
README.md
registry.schema.json
External Libraries
Scratches and Consoles
Extensions
Scratches
CLAUDE.md
This file provides guidance to Claude Code (claude.ai/code) when working with code in this repository.
Build Commands
```bash
# Build registry (validates all agents and outputs to dist/)
uv run --with jsonschema .github/workflows/build_registry.py
# Build without schema validation (if jsonschema not available)
python .github/workflows/build_registry.py
```
Architecture
This is a registry of ACP (Agent Client Protocol) agents and extensions. The structure is:
```
<id>/
├── agent.json      # Agent metadata and distribution info
├── extension.json  # OR extension metadata (same schema as agent.json)
└── icon.svg        # Icon: 16x16 SVG, monochrome with currentColor (optional)
```
Each directory contains either `agent.json` (for agents) or `extension.json` (for extensions), but not both
Build process (`.github/workflows/build_registry.py`):
1. Scans directories for `agent.json` or `extension.json` files
2. Validates against `agent.schema.json` (JSON Schema)
3. Validates icons (16x16 SVG, monochrome with `currentColor`)
4. Aggregates into `dist/registry.json` with separate `agents` and `extensions` arrays
5. Copies icons to `dist/<id>.svg`
CI/CD (`.github/workflows/build-registry.yml`):
- PRs: Runs validation only
- Push to main: Validates, then publishes versioned + `latest` GitHub releases
Validation Rules
- `id`: lowercase, hyphens only, must match directory name
- `version`: semantic versioning (e.g., `1.0.0`)
- `distribution`: at least one of `binary`, `npx`, `uvx`
- `icon.svg`: must be SVG format, 16x16, monochrome using `currentColor` (enables theming)
- **URL validation**: All distribution URLs must be accessible (binary archives, npm/PyPI packages)
Junie
Junie
Junie is a Coding Agent.
It understands the context of the whole project.
This makes it autonomous, and gives it ability to solve big coding tasks.
Type your task here, press
Auto
Brave Mode
Think More
Use Junie From a Single AI Chat
You can now work with Junie and other AI features from a single chat interface.
Try New AI Experience
Learn More
Ask Junie to:
1. Generate project-specific guidelines
2. Write/update a README.md file for this project
3. Based on the last 2-5 git commits, generate a CHANGELOG.md file
4. Develop a "specific feature" using a Test-Driven Development (TDD) approach
Mastering Junie
Junie Getting Started Guide
Coding Guidelines for Your AI Agents
1:1
LF
UTF-8
4 spaces


claude ~/g/zed
zed on main [$] via orbstack via v1.90.0
❯ claude
Claude Code v2.0.27
Sonnet 4.5 · API Usage Billing
/Users/benbrandt/github/zed
> /status
└ Status dialog dismissed
> /help
Claude Code v2.0.27
comman general
custom-com mands
(tab to cycle)
Claude understands your codebase, makes edits with your permission, and executes commands — right from your terminal.
Shortcuts
! for bash mode
/ for commands
@ for file paths
# to memorize
double tap esc to clear input
shift + tab to
ctrl + o for dits verbose output
ctrl + t to show
tab to toggle thinking
shift + ⏎ for newline
ctrl + _ to undo
ctrl + z to suspend
ctrl + v to paste images
For more help: https://docs.claude.com/en/docs/claude-code/overview
Esc to exit
>_ OpenAI Codex (v0.46.0)
model: gpt-5-codex /model to change
directory: ~/github/zed
To get started, describe a task or try one of these commands:
/init - create an AGENTS.md file with instructions for Codex
/status - show current session configuration
/approvals - choose what Codex can do without approval
/model - choose what model and reasoning effort to use
/review - review any changes and find issues
› Summarize recent commits
100% context left · ? for shortcuts
v0.8.0-nightly.20250925.b1da8c21
Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. /help for more information.
> /help
Basics:
Add context: Use @ to specify files for context (e.g., @src/myFile.ts) to target specific files or folders.
Shell mode: Execute shell commands via ! (e.g., !npm run start) or use natural language (e.g. start server).
Commands:
/about - show version info
/auth - change the auth method
/bug - submit a bug report
/chat - Manage conversation history.
list - List saved conversation checkpoints
save - Save the current conversation as a checkpoint. Usage: /chat save <tag>
resume - Resume a conversation from a checkpoint. Usage: /chat resume <tag>
delete - Delete a conversation checkpoint. Usage: /chat delete <tag>
share - Share the current conversation to a markdown or json file. Usage: /chat share <file>
/clear - clear the screen and conversation history

LSP

Use the tools you like
in the editor you love ❤

ACP

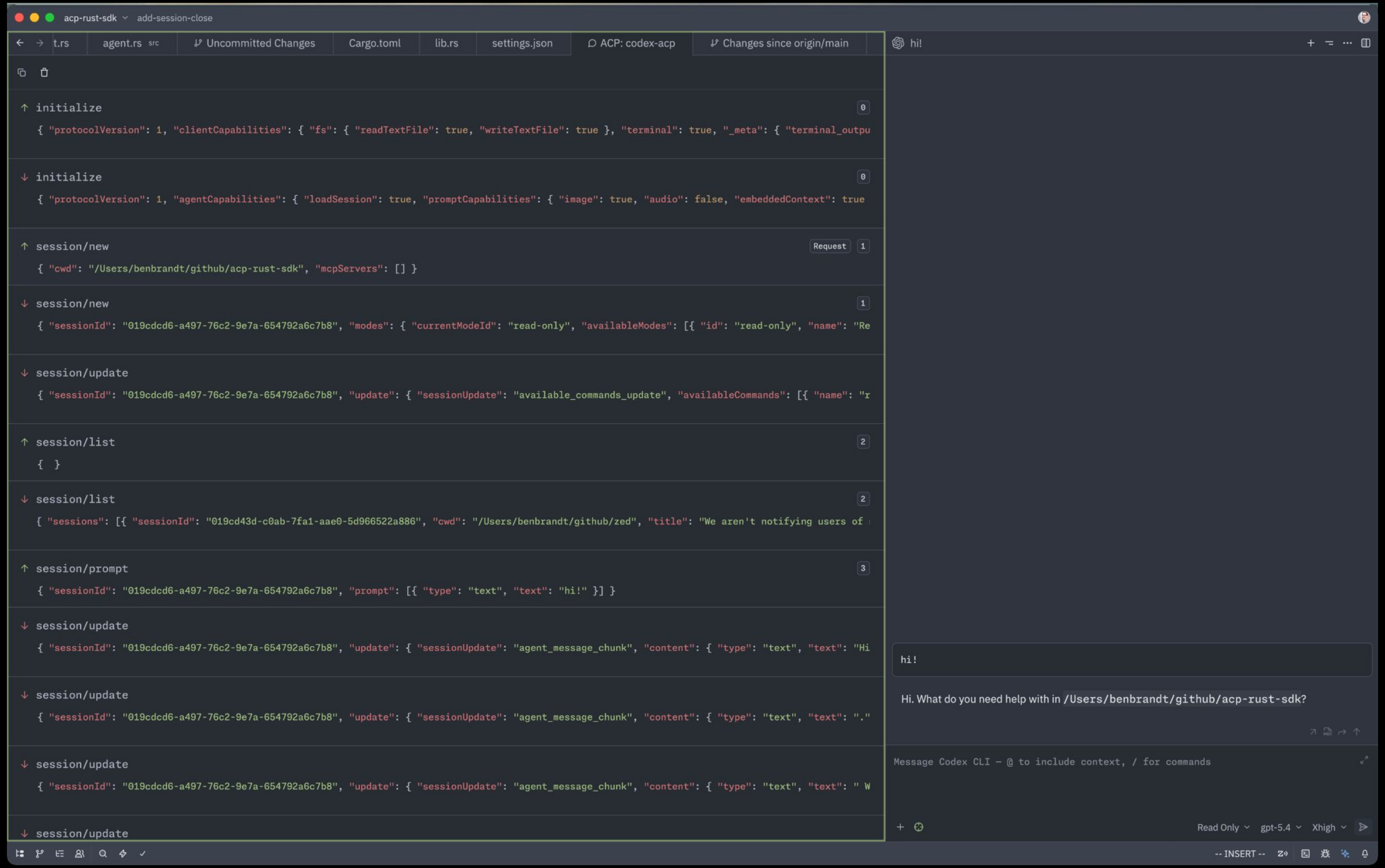
Use the **agents** you like
in the editor you love ❤



Agent/Client Protocol

JSON RPC Spec


```
{ "method":"initialize","params":{"protocolVersion":1,"clientCapabilities":{"fs":{"readTextFile":true
{ "result":{"agentCapabilities":{"loadSession":true,"mcpCapabilities":{"http":true,"sse":false},"pro
{ "method":"session/new","params":{"cwd":"/Users/ignatov/src/intellij-erlang","mcpServers":[]} },
{ "result":{"models":{"availableModels":[],"currentModelId":""},"modes":{"availableModes":[{"descript
{ "method":"session/update","params":{"sessionId":"01de191d-968e-4068-a801-e7e68f1b760f","update":{"a
{ "method":"initialize","params":{"protocolVersion":1,"clientCapabilities":{"fs":{"readTextFile":true
{ "result":{"protocolVersion":1,"agentCapabilities":{"loadSession":true,"resumeSession":true,"session
{ "method":"session/new","params":{"cwd":"/Users/ignatov/src/intellij-erlang","mcpServers":[]} },
{ "result":{"sessionId":"298948c1-41e4-4f46-9497-0ae97d11b8fd","models":{"availableModels":[{"modelId
{ "method":"session/set_mode","params":{"sessionId":"298948c1-41e4-4f46-9497-0ae97d11b8fd","modeId":"
{ "method":"session/update","params":{"sessionId":"298948c1-41e4-4f46-9497-0ae97d11b8fd","update":{"s
{ "method":"session/set_model","params":{"sessionId":"298948c1-41e4-4f46-9497-0ae97d11b8fd","modelId"
{ "method":"session/prompt","params":{"sessionId":"298948c1-41e4-4f46-9497-0ae97d11b8fd","prompt":[{"
{ "method":"session/update","params":{"sessionId":"298948c1-41e4-4f46-9497-0ae97d11b8fd","update":{"s
{ "result":{"stopReason":"end_turn"} }
```


acp-rust-sdk
add-session-close
t.rs
agent.rs src
Uncommitted Changes
Cargo.toml
lib.rs
settings.json
ACP: codex-acp
Changes since origin/main
hi!
initialize
0
{ "protocolVersion": 1, "clientCapabilities": { "fs": { "readTextFile": true, "writeTextFile": true }, "terminal": true, "_meta": { "terminal_outpu
initialize
0
{ "protocolVersion": 1, "agentCapabilities": { "loadSession": true, "promptCapabilities": { "image": true, "audio": false, "embeddedContext": true
session/new
Request
1
{ "cwd": "/Users/benbrandt/github/acp-rust-sdk", "mcpServers": [] }
session/new
1
{ "sessionId": "019cdcd6-a497-76c2-9e7a-654792a6c7b8", "modes": { "currentModeId": "read-only", "availableModes": [{ "id": "read-only", "name": "Re
session/update
{ "sessionId": "019cdcd6-a497-76c2-9e7a-654792a6c7b8", "update": { "sessionUpdate": "available_commands_update", "availableCommands": [{ "name": "r
session/list
2
{ }
session/list
2
{ "sessions": [{ "sessionId": "019cd43d-c0ab-7fa1-aae0-5d966522a886", "cwd": "/Users/benbrandt/github/zed", "title": "We aren't notifying users of
session/prompt
3
{ "sessionId": "019cdcd6-a497-76c2-9e7a-654792a6c7b8", "prompt": [{ "type": "text", "text": "hi!" }] }
session/update
{ "sessionId": "019cdcd6-a497-76c2-9e7a-654792a6c7b8", "update": { "sessionUpdate": "agent_message_chunk", "content": { "type": "text", "text": "Hi
session/update
{ "sessionId": "019cdcd6-a497-76c2-9e7a-654792a6c7b8", "update": { "sessionUpdate": "agent_message_chunk", "content": { "type": "text", "text": "." }
session/update
{ "sessionId": "019cdcd6-a497-76c2-9e7a-654792a6c7b8", "update": { "sessionUpdate": "agent_message_chunk", "content": { "type": "text", "text": " W
session/update
hi!
Hi. What do you need help with in /Users/benbrandt/github/acp-rust-sdk?
Message Codex CLI — @ to include context, / for commands
Read Only
gpt-5.4
Xhigh
-- INSERT --

Current status

Amp
ACP wrapper for Amp - the frontier coding agent

0.7.0

Auggie CLI
Augment Code's powerful software agent, backed by industry-leading context engine

0.18.1

Autohand Code
Autohand Code - AI coding agent powered by Autohand AI

0.2.1

Claude Agent
ACP wrapper for Anthropic's Claude

0.21.0

Cline
Autonomous coding agent CLI - capable of creating/editing files, running commands, using the browser, and more

2.6.1

Codebuddy Code
Tencent Cloud's official intelligent coding tool

2.56.1

Codex CLI
ACP adapter for OpenAI's coding assistant

0.9.5

Corust Agent
Co-building with a seasoned Rust partner.

0.3.7

crow-cli
Minimal ACP Native Coding Agent

0.1.14

Cursor
Cursor's coding agent

0.1.0

DimCode
A coding agent that puts leading models at your command.

0.0.16

Factory Droid
Factory Droid - AI coding agent powered by Factory AI

0.71.0

fast-agent
Code and build agents with comprehensive multi-provider support

0.5.9

Gemini CLI
Google's official CLI for Gemini

0.32.1

GitHub Copilot
GitHub's AI pair programmer

1.0.3

goose
A local, extensible, open source AI agent that automates engineering tasks

1.27.2

Junie
AI Coding Agent by JetBrains

888.180.0

Kilo
The open source coding agent

7.0.41

Kimi CLI
Moonshot AI's coding assistant

1.18.0

Minion Code
An enhanced AI code assistant built on the Minion framework with rich development tools

0.1.39

Mistral Vibe
Mistral's open-source coding assistant

2.4.0

Nova
Nova by Compass AI - a fully-fledged software engineer at your command

1.0.76

OpenCode
The open source coding agent

1.2.24

pi ACP
ACP adapter for pi coding agent

0.0.22

Qoder CLI
AI coding assistant with agentic capabilities

0.1.30

Qwen Code
Alibaba's Qwen coding assistant

0.12.0

Stakpak
Open-source DevOps agent in Rust with enterprise-grade security

0.3.67

30+ Agents



Zed

Native ACP support with real-time editing, syntax highlighting, and agent following.



JetBrains

Full ACP integration coming soon to IntelliJ IDEA, PyCharm, and other JetBrains IDEs.

In Progress

SIDEQUERY

Sidequery

Browser-based development environment with ACP support coming soon.



AionUi

Modern editor with integrated ACP support for seamless agent interaction.



Emacs

Agent-shell.el plugin brings ACP agents to Emacs through a native shell interface.



marimo

Python notebook environment with integrated ACP agent support for data science workflows.

Via Plugin



Neovim

Through two available plugins: CodeCompanion & yetone/avante.nvim.

Via Plugin



Obsidian

Bring your AI agents directly into Obsidian. Chat with Claude Code, Codex, and others from your vault.

AIZEN

aizen

Manage multiple Git branches simultaneously with dedicated terminals and agents in parallel.

DEEPCHAT

DeepChat

A feature-rich open-source AI chat platform supporting multiple cloud and local LLMs.



Tidewave

The coding agent for full-stack web app development. Integrate agents with your web app at every layer.

Via AI SDK Adapter

WEB BROWSER

Web Browser

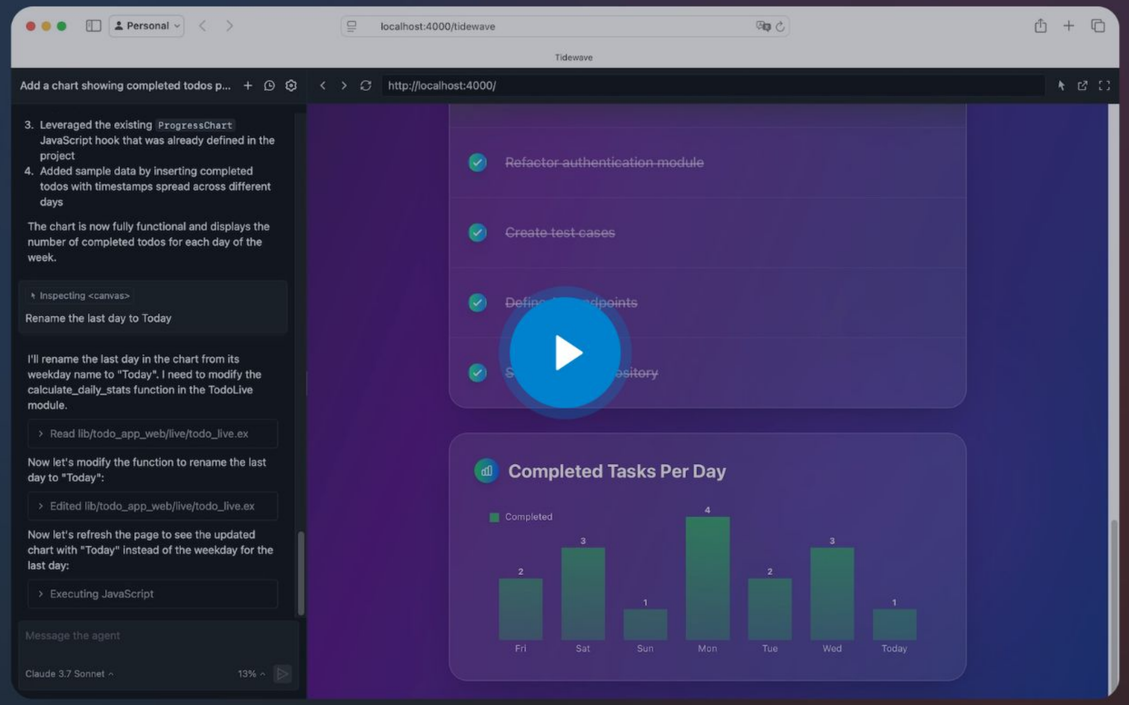
AI SDK-powered browser client with ACP support via @mcpc/acp-ai-provider.

40+ Clients

Your coding agent knows your code. Now it knows your web app too.

Connect Claude Code, Codex, or Copilot to your running app: database, logs, and live UI. Point to any element to reference it. Your agent sees what you see.

Install Tidewave Join our Discord



Search... ⌘K

Get Started
- Introduction
- Architecture
- Agents
- Clients
- ACP Registry

Protocol
- Overview
- Initialization
- Session Setup
- Session List
- Prompt Turn
- Content
- Tool Calls
- File System
- Terminals
- Agent Plan
- Session Modes
- Session Config Options

Get Started

ACP Registry

Copy page

The easiest way to find and install ACP-compatible agents.

Overview

The ACP Registry is an easy way for developers to distribute their ACP-compatible agents to any client that speaks the protocol.

This is a curated set of agents, including only the ones that **support authentication**.

Visit **the registry repository on GitHub** to learn more about it.

Available Agents

Amp

ACP wrapper for Amp - the frontier coding agent

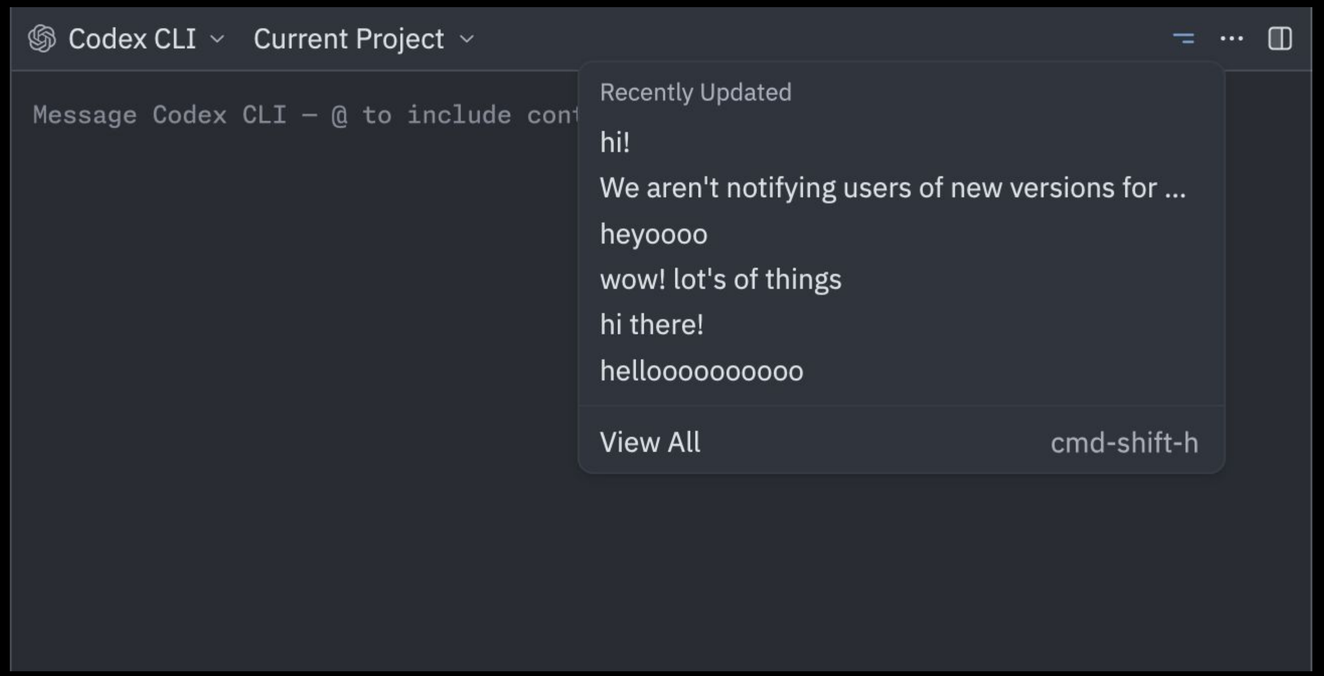
`0.7.0`

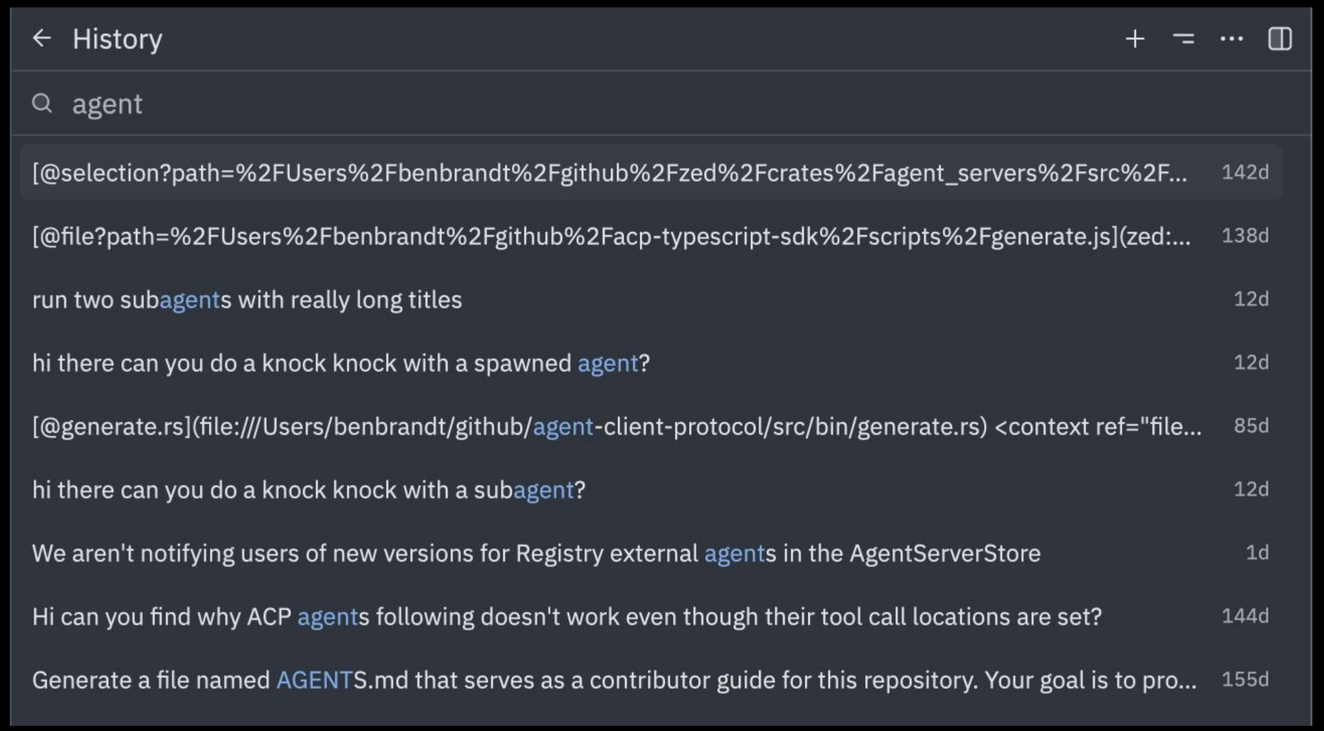
Auggie CLI

Augment Code's powerful software agent, backed by industry-leading context engine

`0.18.1`

Session History





The vision

Clients:
More choice for your users,
without custom integrations.

Agents:
Distribute your agent where your users do their work.

Developers:
Use the best agent for the job
with tools you already love.

The future

Work in progress, short-term

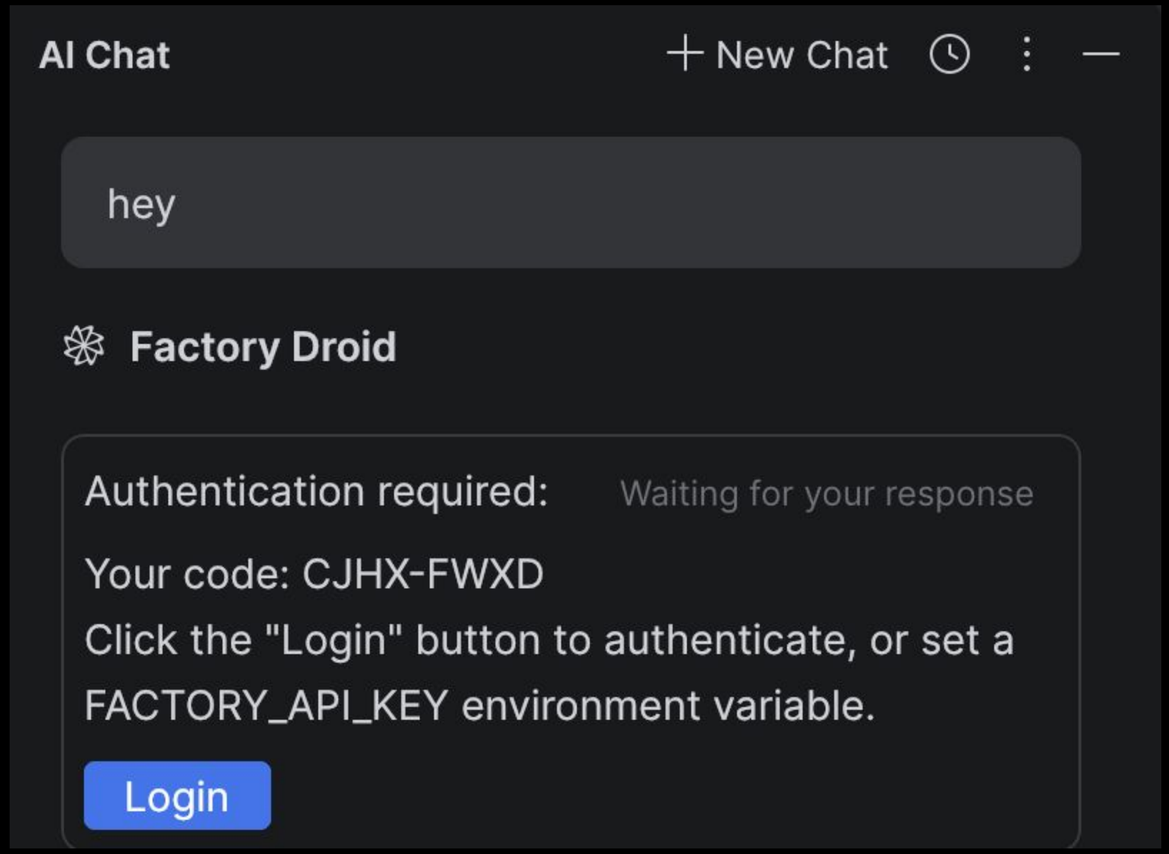
Auth support

Interactive login

Asking from ENV

Terminal as a backup plan

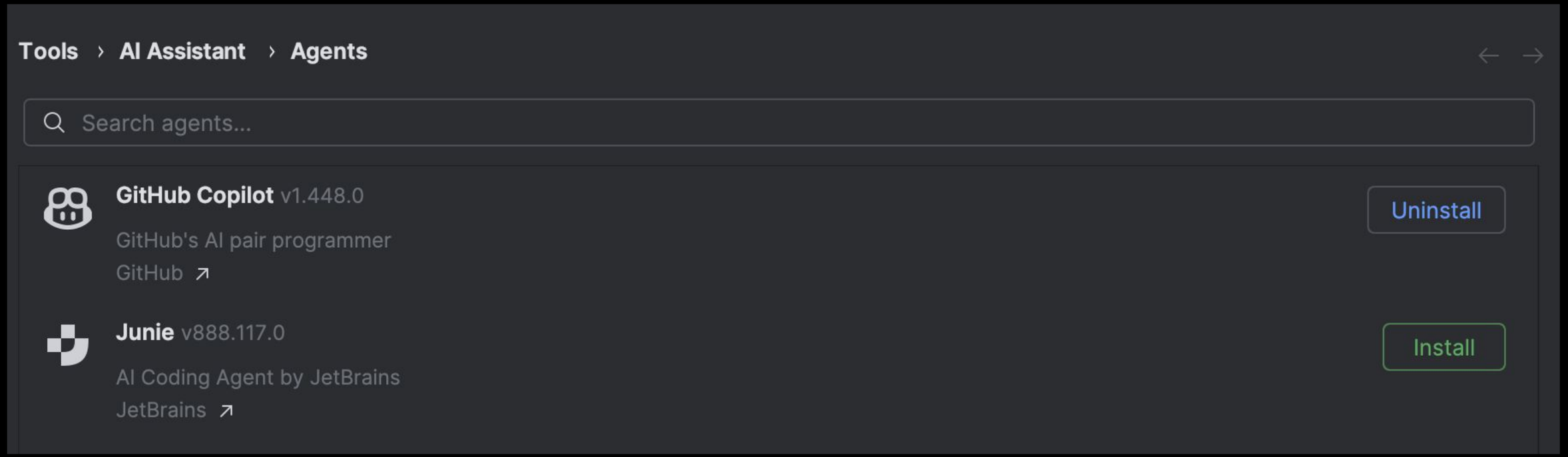
OAuth 2.0



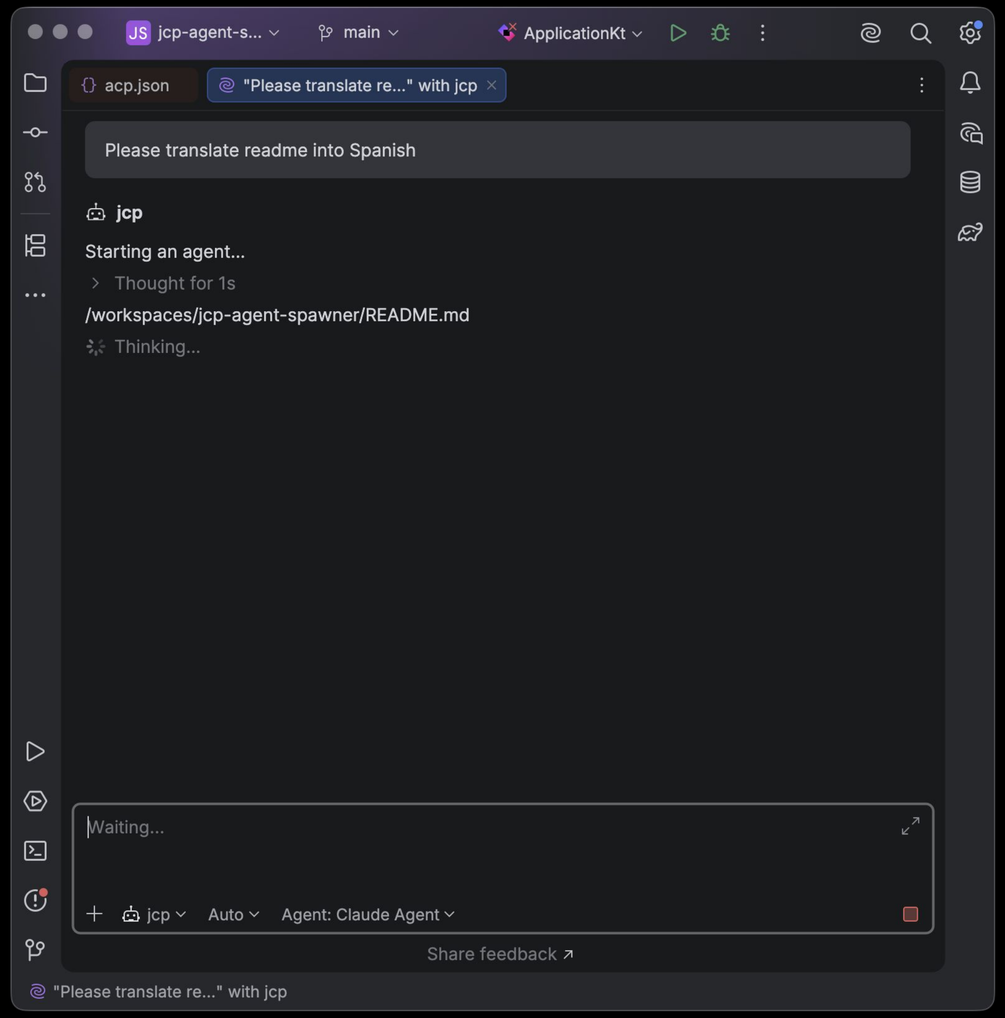
Next Edit Suggestion

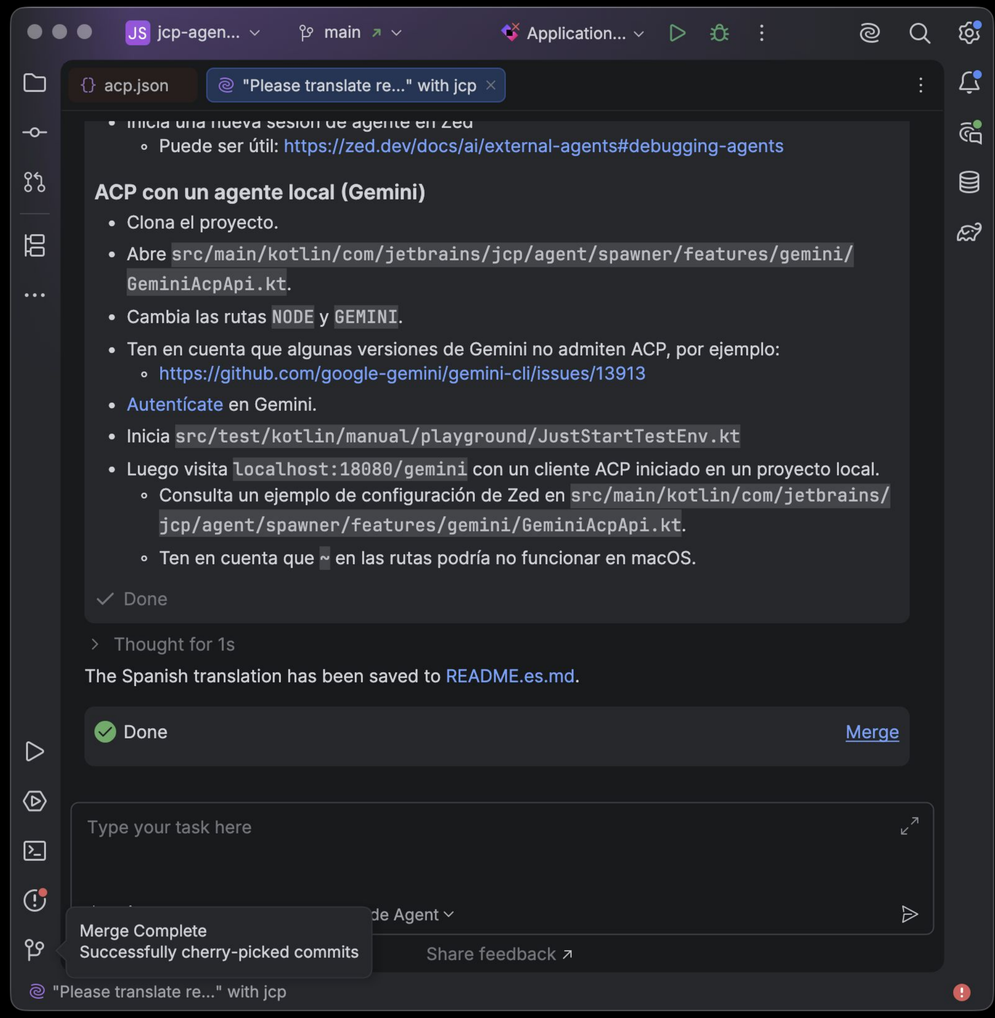


Enterprise registry



Remote agents





Keeping on par with TUI

MCP-style elicitation

Agenting-driven tasks follow ups

Skills completion & suggestion

Session management

delete/close

store/restore

fork/join

Async communication: steering & queuing

Telemetry

MCP over ACP

What can we build on top of standard communication patterns with agents?

Spare parts

The registry

SDKs

TS / Rust / Kotlin / Java / Python
Go / Swift / Dart / .Net

Agentic frameworks

LangChain with deepagents
Koog
LlamaIndex
Thinkwell

Chat connectors

Slack

Discord

Telegram

Product examples

00:40

Search

Agmente

Agent Client Protocol client

Open

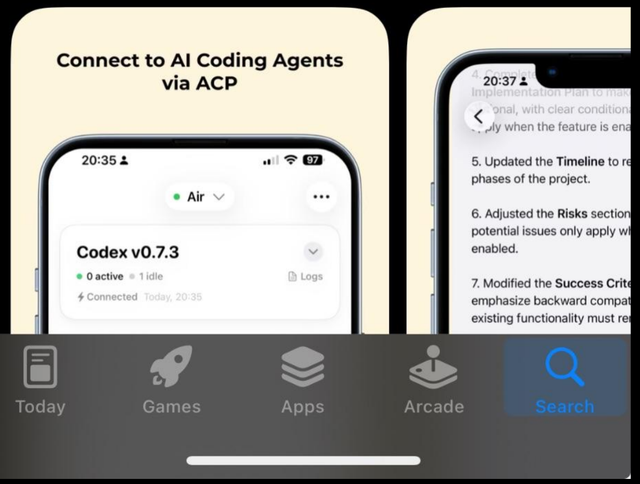
AGE	CATEGORY	DEVELOPER	L/
4+	Developer Tools	Peng Lyu	
Years Old			

What's New

Version 1.2 3w ago

- Faster and more responsive session switching
- Better stability when managing multiple agent connections more

Preview



00:34

Qwen

Search sessions

Qwen Code v0.8.1

0 active 2 idle Logs

Connected Today, 00:34

TODAY

hey
The project appears to be an agent-cli...

hey
Hello! How can I assist you with the A...

+ New Session

00:34

Desc the project

The project appears to be an agent-client-protocol implementation, which is a Rust/Cargo-based project with a focus on defining and managing communication protocols between agents and clients. It includes a `Cargo.toml` file, suggesting it's a Rust crate, and a `package.json` file, indicating it might also have JavaScript/TypeScript components. The presence of `.release-plz.toml` suggests automated release management. The project also includes documentation files like `AGENTS.md`, `CHANGELOG.md`, `CONTRIBUTING.md`, and `README.md`, which provide insights into the project's purpose, contribution guidelines, and changelog. The directory structure indicates a focus on protocol definition and implementation, possibly for AI agents or similar systems.

Default

Message the agent...

- richer per-status summaries instead of only `supported/authRequired/methodNotFound/other`
- `protocolSignatures` for `session/new` and the post-init methods
- `consistencyChecks` such as:
  - `advertisedButMissing`
  - `worksButNotAdvertised`
  - `authBoundaryChanged`
  - `responseShapeChanged`
- a `Protocol Drift Warnings` section in the markdown report

If you want to go further without auth
There are two additional options:

- **agent test mode**: a special env var that enables fake authenticated behavior in CI
- **reference/mock agents**: dedicated conformance targets to validate the probe engine itself

That still is not real auth, but it does let you validate post-auth protocol shape in a controlled way.

kotlin-web-site air/search-config-2546 / Search config housekeeping

Search tasks

New Task

Running

Search config housekeeping
Running quick format and l... +34 -0

Add search suggestions drop...
Extend API for partial queri... +12 -0

Implement GET /search handler
Inspecting existing handlers...

Done

Build SearchBox component
Done +242 -0

Add /api/demo-search
Done +39 -0

Perform a search config housekeeping — keep existing indexes and add commented demo section for local testing

I'll help you perform housekeeping on the search-config.json file and add a commented demo section for local testing. Let me first examine the current config structure and then create a clean version with a demo section.

Read {} search-config.json

Perfect! Now I have the complete search-config.json structure. Let me update my todo and create the cleaned up version with proper housekeeping and a demo section.

{} search-config.json +11 -0

Next, I'll make sure the new section doesn't break JSON parsing when comments are stripped. For that, I'll generate a quick Node validation script.

TS validate-search-config.ts +23 -0

Now let me run the script to confirm the config parses correctly:

Successfully validated

Config looks good — now I'll run format and lint checks to confirm everything meets style rules.

PLAN 3/4

- Read and parse current config structure
- Add commented demo section at the bottom
- Validate JSON after removing comments
- Run quick format and lint checks

Follow-up on this task, @ for mentions, / for commands

Sonnet 4 Ask Permission

← **Post**



Peter Steinberger 🦞  

@steipete

acp subagents are lit

7:08 pm · 7 Mar 2026 · **44.5K** Views

Research

https://symposium.dev

niquola/agentic-workspace

github.com

agentclientprotocol

Type / to search

Overview | Repositories 11 | Discussions | Projects | Packages | Teams 7 | People 15 | Insights | Settings

Agent Client Protocol (ACP)

A protocol for connecting any editor to any agent

Verified

127 followers | https://agentclientprotocol.com/

Follow

README.md



Agent Client Protocol

The Agent Client Protocol (ACP) standardizes communication between *code editors* (interactive programs for viewing and editing source code) and *coding agents* (programs that use generative AI to autonomously modify code).

Learn more at agentclientprotocol.com.

Integrations

- Schema
- Agents
- Clients
- Official Libraries
  - **Kotlin**: `acp-kotlin` - Supports JVM, other targets are in progress
  - **Python**: `python-sdk` - See examples
  - **Rust**: `agent-client-protocol` - See examples/agent.rs and examples/client.rs
  - **TypeScript**: `@agentclientprotocol/sdk` - See examples/
- Community Libraries

Pinned

Customize pins

agent-client-protocol Public
A protocol for connecting any editor to any agent
Rust 1.8k 138

kotlin-sdk Public
Agent Client Protocol Kotlin SDK
Kotlin 38 9

python-sdk Public
Python SDK for ACP clients and agents.
Python 135 12

rust-sdk Public
Rust SDK for ACP clients and agents.
Rust 46 10

View as: Public

You are viewing the README and pinned repositories as a public user.

Get started with tasks that most successful organizations complete.

Top discussions this past month

- **Support for questions and "wizards" like AskUserQuestionTool** ↑ 4 💬 4
- **Remote ACP Agents** ↑ 2 💬 0
- **How to "always approve" tool calls for specific MCP server** ↑ 1 💬 0

View all discussions

People

Invite someone

Top languages

Rust MDX Java Kotlin TypeScript

Most used topics Manage

acp ai agent-client-protocol agentclientprotocol agents

Requests for Dialog (RFDs)

Copy page

Our process for introducing changes to the protocol

A “Request for Dialog” (RFD) is ACP’s version of the RFC process. RFDs are the primary mechanism for proposing new features, collecting community input on an issue, and documenting design decisions.

When to write an RFD

You should consider writing an RFD if you intend to make a “substantial” change to ACP or its documentation. What constitutes a “substantial” change is evolving based on community norms and varies depending on what part of the ecosystem you are proposing to change.

Some changes do not require an RFD:

- Rephrasing, reorganizing or refactoring
- Addition or removal of warnings
- Additions that strictly improve objective, numerical quality criteria (speedup, better browser support)
- Fixing objectively incorrect behavior

The RFD Process

1. Propose by opening a PR

Fork the repo and copy `docs/rfds/TEMPLATE.md` to `docs/rfds/my-feature.md` (using kebab-case naming). The RFD can start minimal -just an elevator pitch and status quo are enough to begin dialog. Pull requests