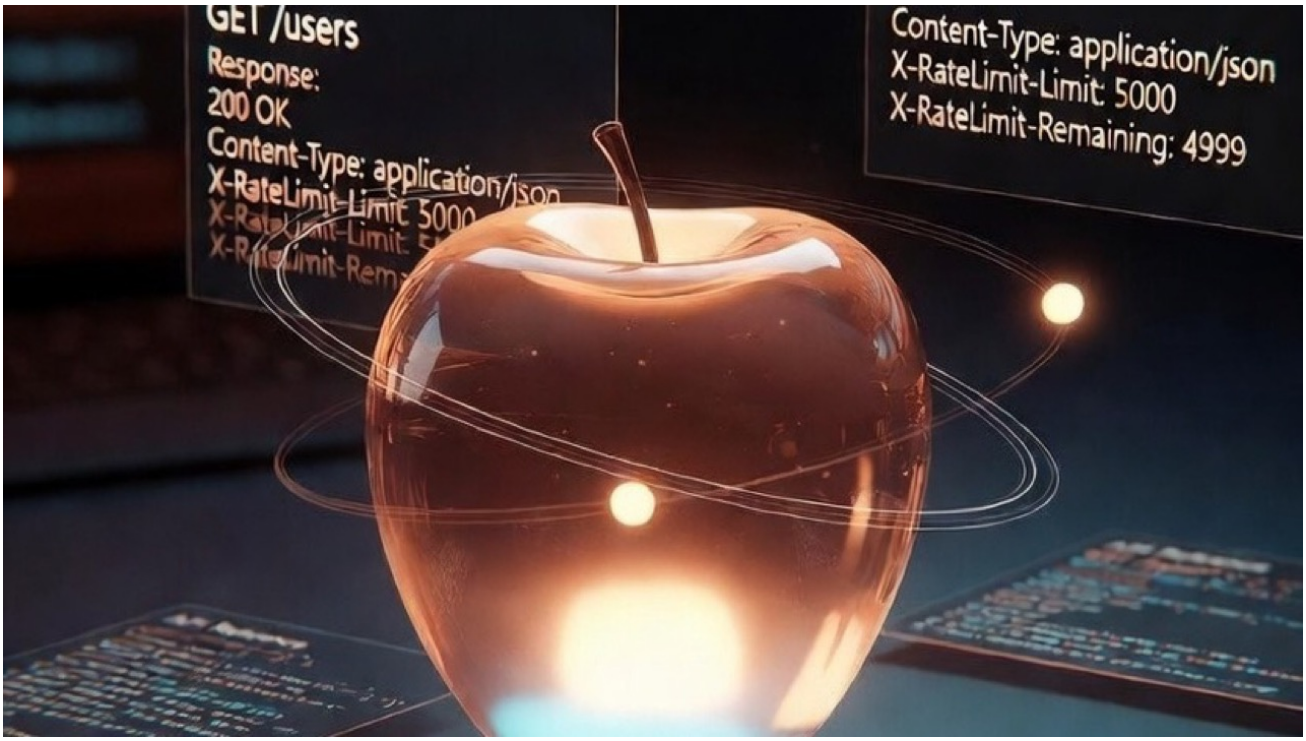

Cupertino v0.10: Full Coverage and Community Contributions

TL;DR: All 307 Apple frameworks are now indexed. 302,424 documentation pages searchable. Community contributors added Agent Skill support and a Claude Code plugin. Homebrew resource bundle fix.

The Completionist Achievement: Unlocked

I wanted every Apple framework in the index. Not most of them. *All* of them.

Turns out, the crawler was only finding ~20 frameworks by following links from Apple's developer homepage. That's like exploring a library by only reading the "Staff Picks" shelf. The rest of Apple's documentation was sitting there, patiently waiting to be discovered.

The fix: seed the crawler from Apple's own `technologies.json` -- their internal index of every framework. Feed that to the BFS crawler, and suddenly the whole library opens up.

But that was just the first of five bugs hiding in the crawl engine:

Bug	What Went Wrong	Fix
Session resume ignored `startURL`	Resumed any session, even for wrong URLs	Check URL match before resuming
BFS-only discovery	Homepage links only find ~20 frameworks	Seed from `technologies.json`
Case-sensitive URL matching	Apple's JSON returns mixed-case paths	`shouldVisit()` now case-insensitive
0.5s request delay	Life is short	Reduced to 0.05s

Bug	What Went Wrong	Fix
Skip without enqueue	Content-unchanged pages never enqueued their links	Moved link enqueue before skip check

That last one was sneaky. If a page's content hadn't changed since the last crawl, the crawler skipped it -- but skipped *before* adding its child links to the queue. Entire subtrees silently disappeared.

The result: 13 previously missing frameworks crawled and indexed. CoreGraphics, CoreMedia, CoreVideo, CoreText, CoreAudio, dnssd, Hypervisor, IOBluetooth, HIDDriverKit, Matter, OpenGLES, USBDriverKit, and CoreAudioTypes.

Yes, even OpenGLES. Deprecated, but completionism has no room for judgment.

Framework	Documents
Kernel	39,396
Matter	24,320
Swift	17,466
AppKit	12,443
Foundation	12,423
UIKit	11,158
SwiftUI	7,062
IOBluetooth	3,895
CoreGraphics	3,500+
CoreMedia	2,800+
...	...
307 Frameworks	**302,424**

Your AI agent can now answer questions about Hypervisor's vCPU mapping *and* IOBluetooth's RFCOMM channels. You're welcome.

Community Contributions

This release marks a milestone I'm genuinely excited about: the first external contributions to Cupertino.

Both add new ways to use Apple documentation without running an MCP server. Two contributors saw something they wanted, built it, and sent a PR. That's the open source dream right there.

Agent Skill Support (PR #167)

Tijs Teulings added support for using Cupertino as a stateless Agent Skill. AI coding agents get direct access to Apple documentation via CLI commands with JSON output -- no MCP server process needed.

Install with OpenSkills:

```
npx openskills install mihaelamj/cupertino
```

Or manually copy the skill definition to `.claude/skills/cupertino/`.

No daemon. No stdio pipe. Just `cupertino search "SwiftUI" --format json` and you're done.

Claude Code Plugin (PR #169)

Gustavo Ambrozio took Tijs's skill and leveled it up into a proper Claude Code plugin with marketplace support.

```
claude /plugin marketplace add https://codeberg.org/CupertinoHQ/cupertino.git
```

The plugin adds a manifest, marketplace metadata, and improved skill descriptions with better trigger phrases. One command to install, zero configuration after that.

Thank you Tijs and Gustavo. It's a great feeling to see the project grow beyond a single-maintainer effort.

The Homebrew Symlink Saga

Here's a fun one. After the v0.10 release, `brew install cupertino` worked fine... until you actually ran any command. Instant crash:

```
Fatal error: could not load resource bundle:
from /opt/homebrew/bin/Cupertino_Resources.bundle
```

The problem: Homebrew symlinks *files* from the Cellar to `/opt/homebrew/bin/`, but not *directories*. The resource bundle was sitting happily in the Cellar, but `Bundle.main.bundleURL` (which doesn't resolve symlinks) was looking in `/opt/homebrew/bin/` where the symlink lives. Two ships passing in the night.

The fix was two-pronged:

1. **Formula**: Added a `post_install` hook that manually creates the symlink for the resource bundle directory
2. **Code**: `CupertinoResources.bundle` now resolves symlinks via `executableURL.resolvingSymlinksInPath()` before falling back to `Bundle.module`

Belt and suspenders. If you had the crash, `brew update && brew upgrade cupertinohq/tap/cupertino` fixes it.

Search Improvements

Framework search now includes synonym matching. Because nobody remembers if it's "CoreGraphics" or "Core Graphics":

You search for	Finds
"Core Graphics"	coregraphics
"UIKit"	uikit
"SwiftUI"	swiftui
"Core Data"	coredata

Case-insensitive, space-tolerant. Type it however you remember it.

Install or Update

```
# New install
brew tap cupertinohq/tap https://codeberg.org/CupertinoHQ/homebrew-tap.git
brew install cupertinohq/tap/cupertino

# Update existing
brew update && brew upgrade cupertinohq/tap/cupertino

# Or one-line install
bash <(curl -sSL
https://codeberg.org/CupertinoHQ/cupertino/raw/branch/main/install.sh)
```

Then:

```
cupertino setup  # Download databases (~30 seconds)
cupertino serve  # Start MCP server
```

Issues and PRs Closed

- #159 - Missing framework documentation
- #160 - Crawler improvements for full coverage
- #167 - Agent Skill support (@tijs)
- #169 - Claude Code plugin (@gpambrozio)
- #162 - CONTRIBUTING.md (@William-Laverty)
- #133 - README link fix (@lwdupont)
- #172 - Crawler session resume and delay fixes

Links

- Codeberg Repository
- Documentation
- Tag v0.10.0