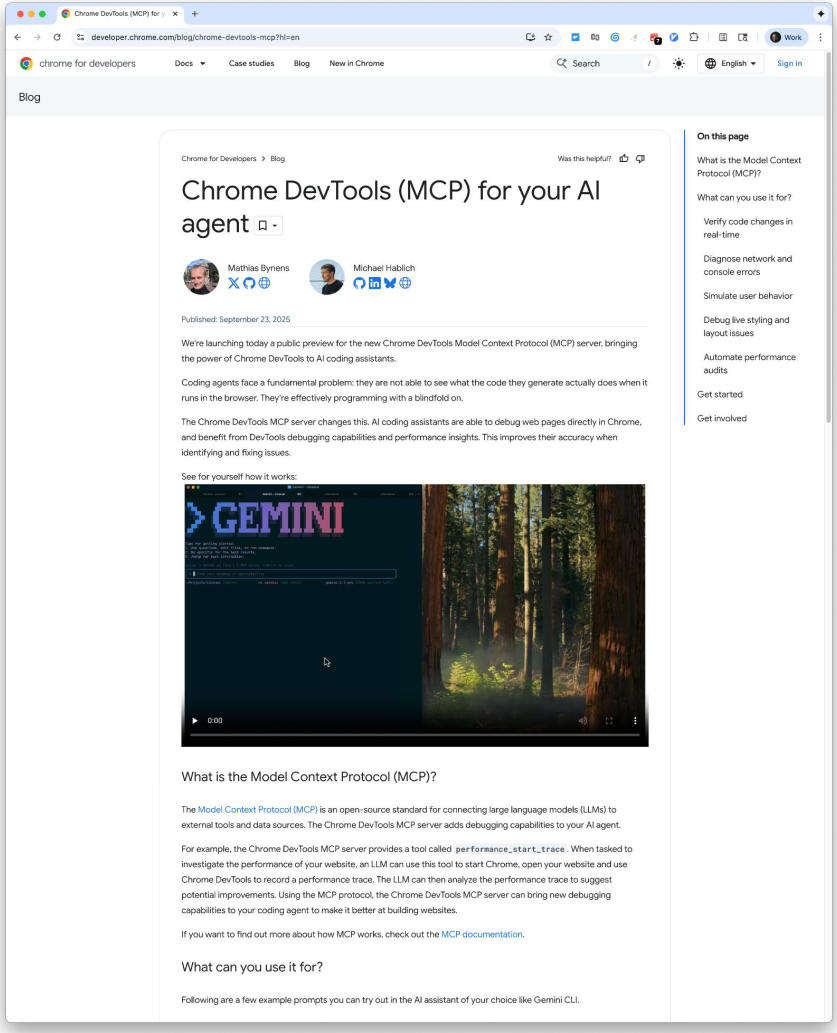
Building the Chrome DevTools MCP server

Jack Franklin, MCP Summit, October 2025



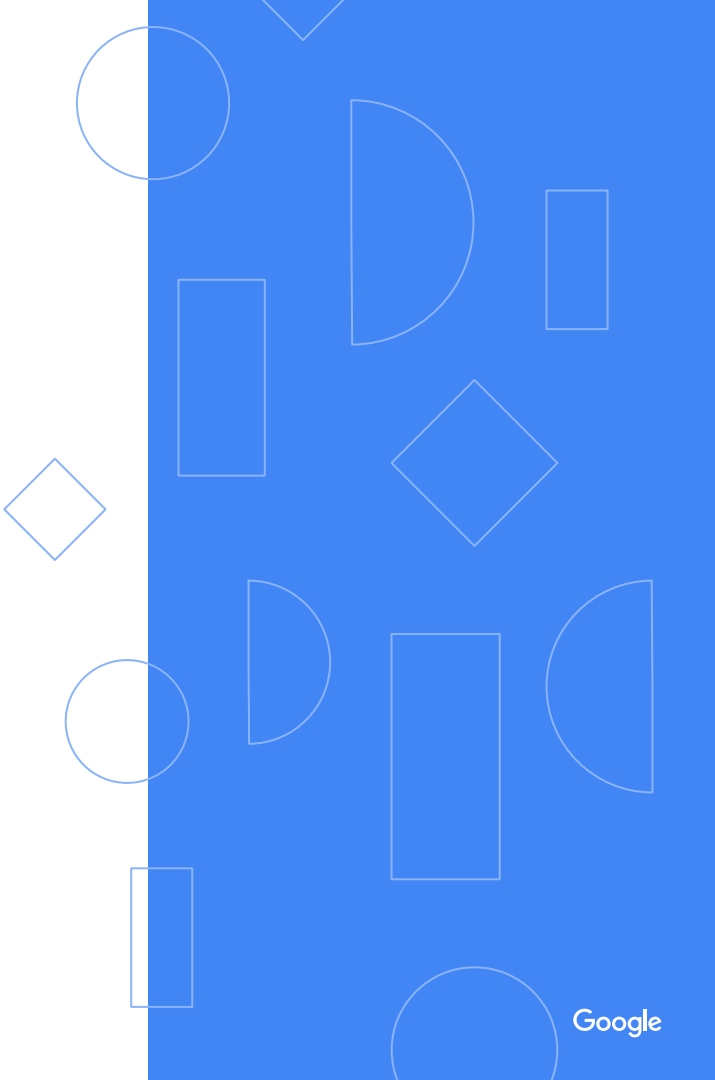
https://developer.chrome.com/blog/chrome-devtools-mcp

Building for AI users

Learning & Abstractions

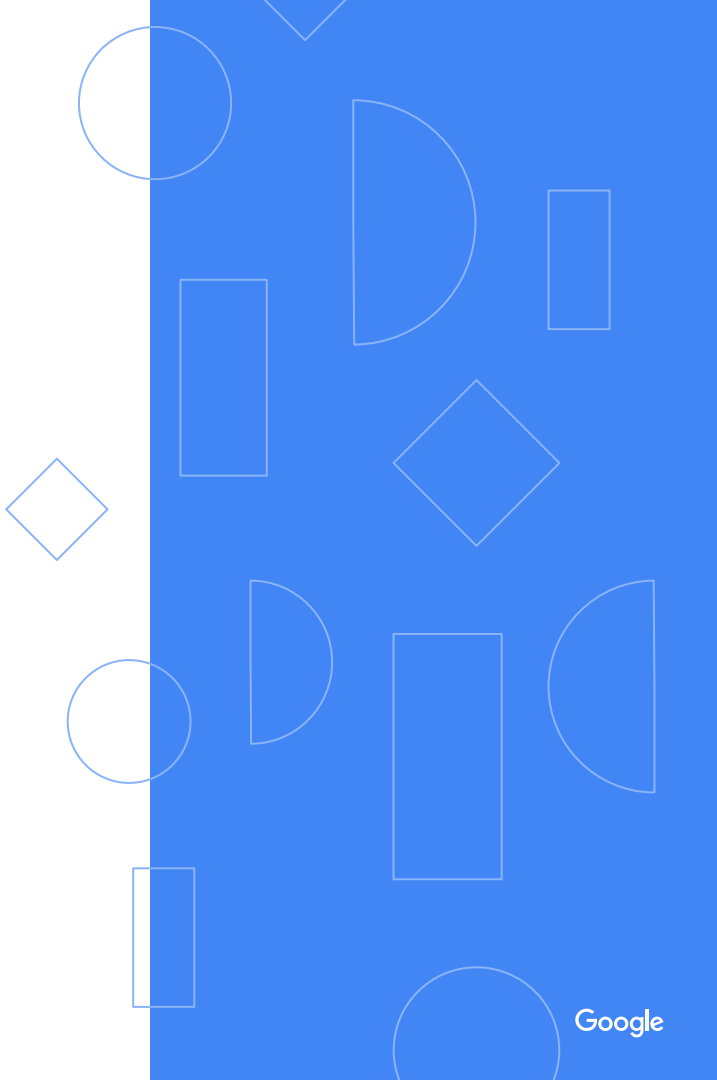
Challenges

Building for AI users



Google

What is our value proposition that an LLM can’t do on its own?



Google

Verify in the browser that your change works as expected

Some images on localhost:8080 are not loading. What's happening?

Why does submitting the form fail after entering an email address?

---

localhost:8080 is loading slowly. Make it load faster.

Verify in the browser that your change works as expected

Some images on localhost:8080 are not loading. What's happening?

Why does submitting the form fail after entering an email address?

Can we enable an AI to answer these prompts well?

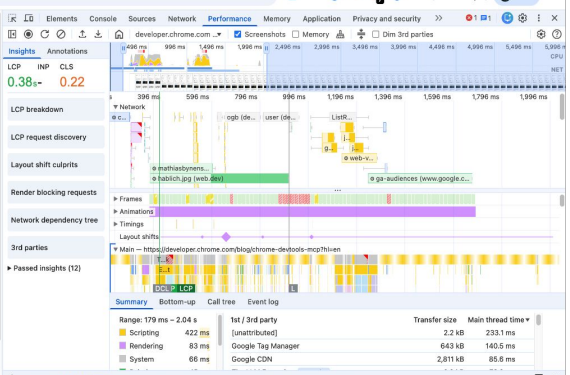
What can DevTools give AI agents that no one else can?

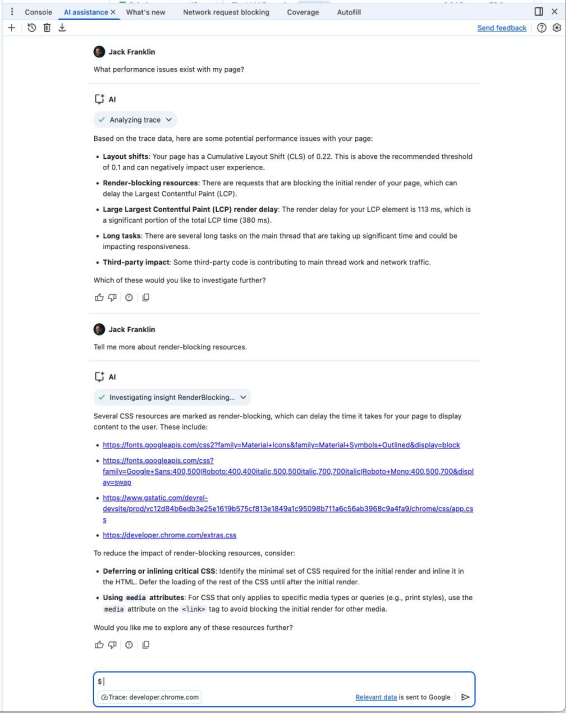
Runtime data

Relating source code to production

Enhanced debugging capabilities

Learning from AI in DevTools





https://developer.chrome.com/docs/devtools/ai-assistance

#1

Prompt engineering

#2

How LLMs use function calls

#3

Context window engineering

Representing call trees

```
# All URL #s:
  * 0: https://www.example.com/spa/123.hash-7363f4688350186a92b4.js
  * 1: https://www.example.com/spa/hash-8d85eb1943d61b172196.js
  * 2: https://www.example.com/spa/124.hash-285098f1d43ad35851b6.js
  * 3: https://www.example.com/ruxitagentjs_A7NVfghqrtux_10291240606133530.js
  * 4: https://www.example.com/kfrcdn/vice_loade
  * 5: https://www.example.com/spa/11.hash-262d4
  * 6: https://www.example.com/spa/2.hash-a8a5bf
  * 7: https://www.example.com/spa/1.hash-d8f430
  * 8: https://www.example.com/spa/30.hash-30736

# Call tree:
Node: 1 – Task
dur: 541.4
self: 0.3
Children:
  * 2 – Evaluate script

Node: 2 – Evaluate script
dur: 541.1
self: 1.7
URL #: 0
Children:
  * 3 – Compile script
  * 4 – (anonymous)
  * 5 – Run microtasks

Node: 3 – Compile script
dur: 29.3
self: 29.3
URL #: 0

Node: 4 – (anonymous)
```

```
# All URL #s:
* 0: https://www.gstatic.com/firebasejs/6.6.1/firebase-performance.js

# Call tree:
1;Task;0.9;0;;2;S
2;Timer fired;0.9;0;;3
3;Function call;0.9;0.1;0;4
4;(anonymous);0.8;;0;5
5;(anonymous);0.8;;0;6-8
6;Ot.getEntriesByType;0.1;;0;8
7;le.createOobTrace;0.6;0.2;0;9-11
8;getEntriesByType;0.1;0.1;;
9;le;0.1;0.1;0;
10;ie;0.2;;0;11-13
11;Ot.requiredApisAvailable;0.2;0.2;0;
12;oe;0;;0;13
13;setTimeout;0;0;;`;
```

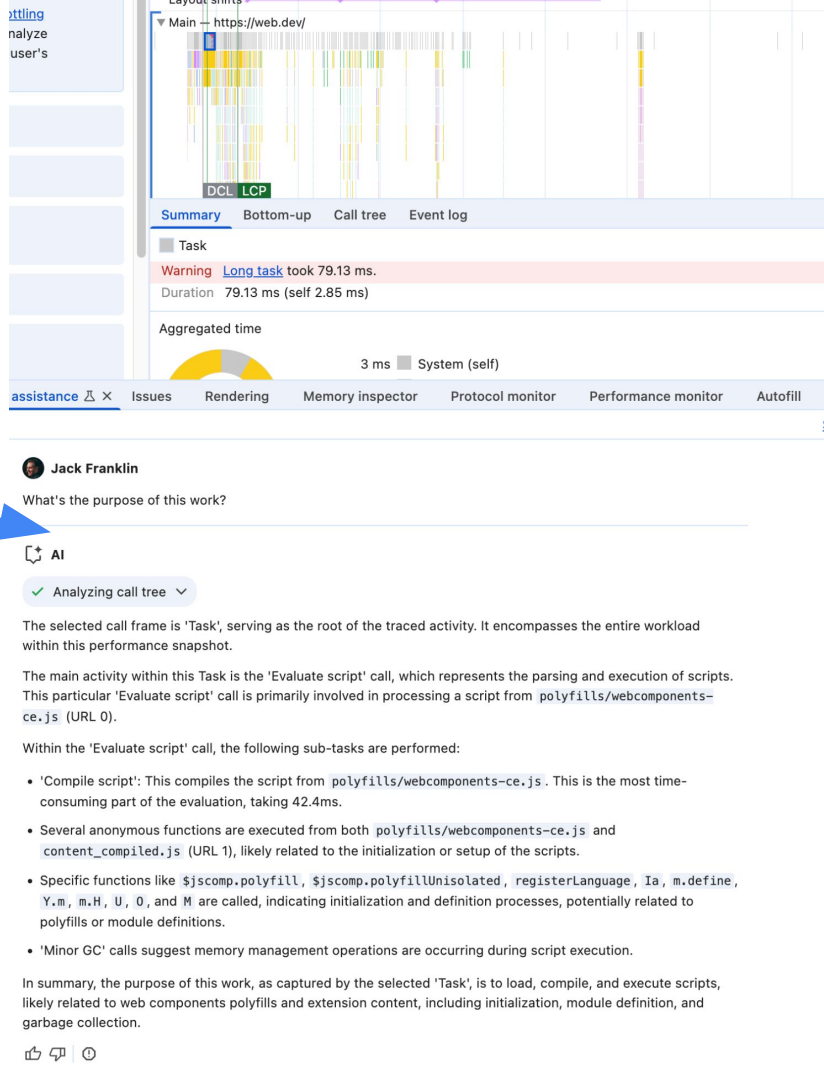
Finding the right abstraction level

High level, powerful tools, or low level building blocks?

Our first MCP server exposed high level tools

```
debugPerformanceProblem({
  url: 'www.google.com',
  prompt: 'why does the page load slowly'
});

debugStylingProblem({
  url: 'www.google.com',
  prompt: 'why does the page load slowly',
  selector: 'h1.main-heading'
});
```

```
debugPerformanceProblem({
  url: 'www.google.com',
  prompt: 'why does the page load slowly'
});
```

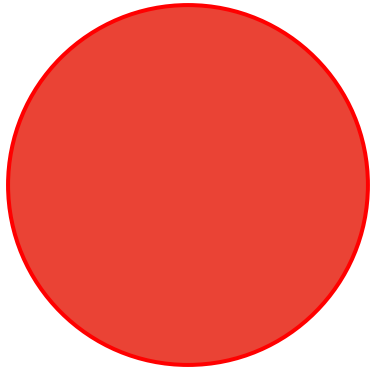


But...we didn't find it provided the value we wanted

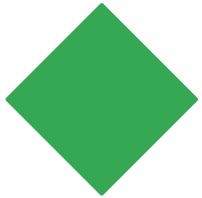
Too niche and not composable

Complex setup process

Hard when the AI client can't get at the browser

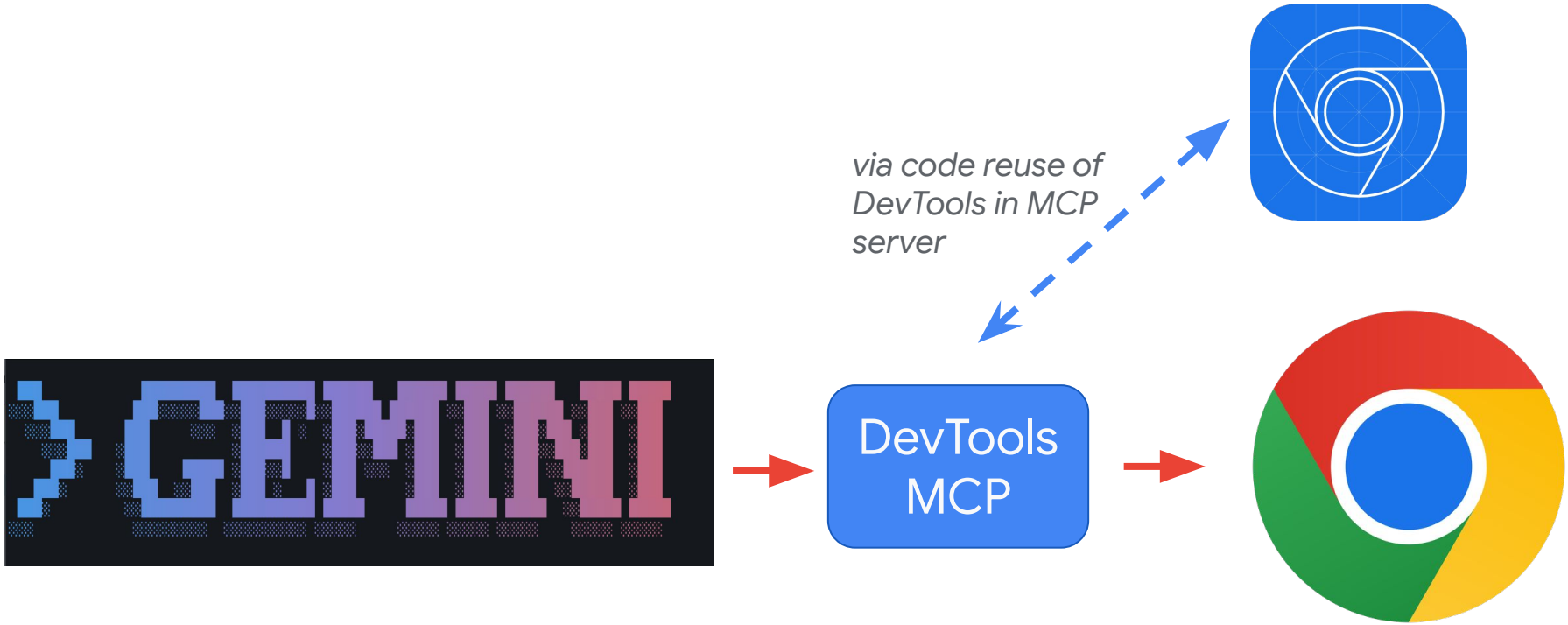


V2: expose lower level building blocks



We focused on tools to replicate common manual user actions when debugging

```
click, fill, fill_form, hover, upload_file

navigate_page, new_page, wait_for

emulate_cpu, emulate_network, resize_page

performance_start_trace,
performance_stop_trace

get_network_request, list_network_requests

evaluate_script, take_screenshot,
```



Now DevTools can talk to a regular Chrome, with no extra setup required.

The challenges of building MCP servers and working with AI

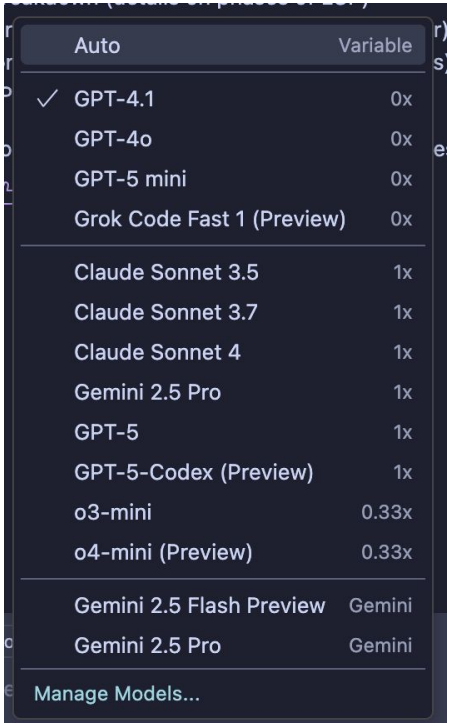
The amount of different environments

Google

The matrix of clients & models is vast.

VSCode
Cursor
Cline
WebStorm
Zed

(and model specific options like Claude Code, Gemini CLI)



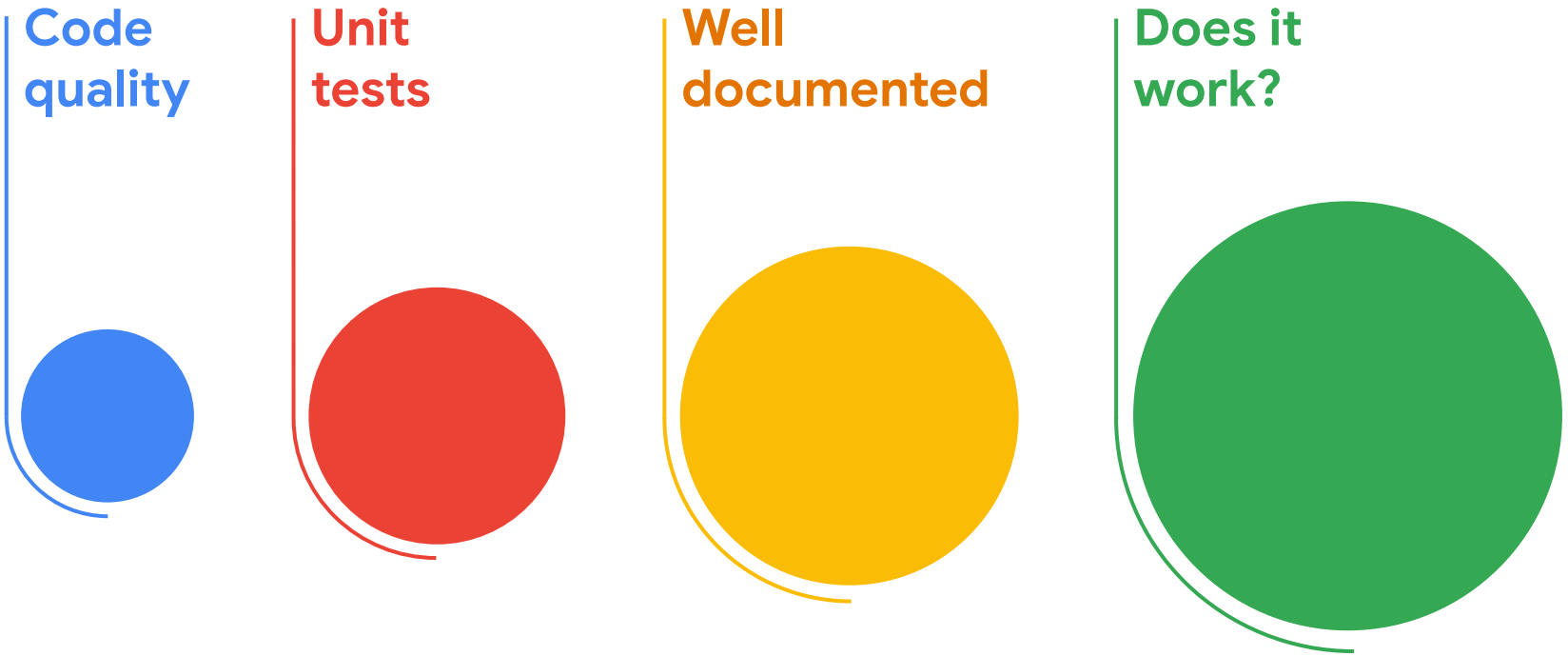
Variance in user experience


“Client X doesn’t call the performance tracing functions in the right order”
“Client Y doesn’t call the performance tracing functions in the right order”
Great, I’ve landed PR#24 that updates the function description to remove the confusion.

Evaluating our changes

Google

PR: “Improve function tool descriptions for Performance tools”

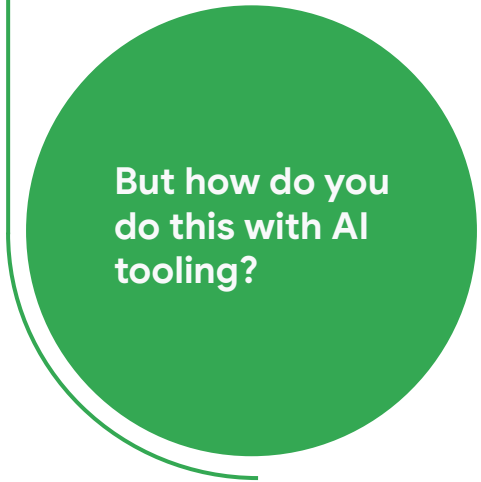


Code quality

Unit tests

Well documented

Does it work?



Good dataset

Representative set of websites with a variety of problems that DevTools can fix.

Effortless collection

Make it really easy to make a change and gather output from an AI interaction with DevTools.

Evaluation framework

Using “LLM as judge” and another techniques to score quality of output.

(index)	id	average	median	allScores
0	'lcp_render_delay/v1'	'2.10'	'2.00'	'-6, -1, -1, -1, 2, 2, 5, 5, 8, 8'
1	'lcp_render_delay/v2'	'1.22'	'-1.00'	'-3, -2, -1, -1, -1, 1, 2, 8, 8'
2	'lcp_render_delay/v3'	'3.33'	'4.00'	'-1, 4, 7'
3	'lcp_render_delay/v4'	'6.63'	'8.00'	'2, 2, 2, 5, 5, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8'
4	'lcp_render_delay/v5'	'7.00'	'8.00'	'2, 5, 8, 8, 8, 8, 8, 8, 8'
5	'lcp_render_delay/v6'	'6.00'	'8.00'	'-1, 2, 5, 5, 5, 8, 8, 8, 8, 8, 8, 8'

Supporting richer output

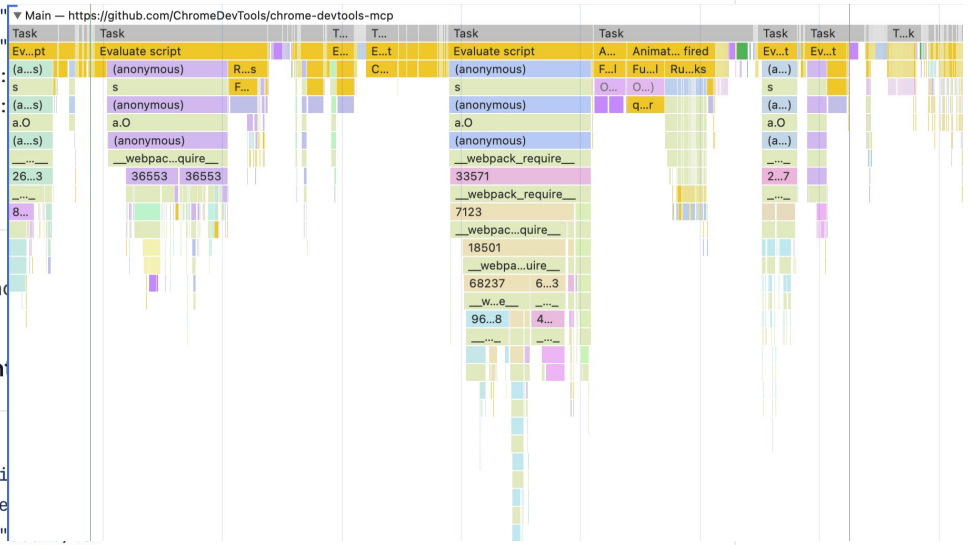
Google

5.2.1 Text Content

```
{
  "type": "text",
  "text": "Tool result text"
}
```

5.2.2 Image Content

```
{
  "type": "image",
  "data": "base64-encoded-data",
  "mimeType": "
  "annotations"
    "audience":
    "priority":
  }

}
```

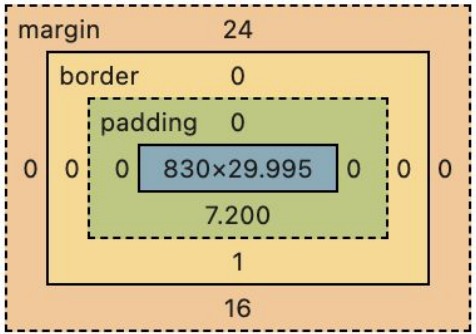


This example demo

5.2.3 Audio Con

```
{
  "type": "audi
  "data": "base
  "mimeType": "
}
```

Seems like a good fit for web debugging
...but what if the user is in a terminal client and cannot view images?



The pace of change

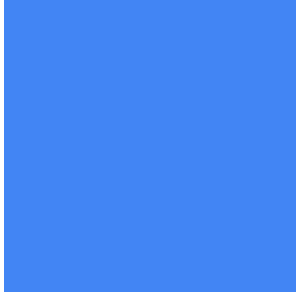
Google

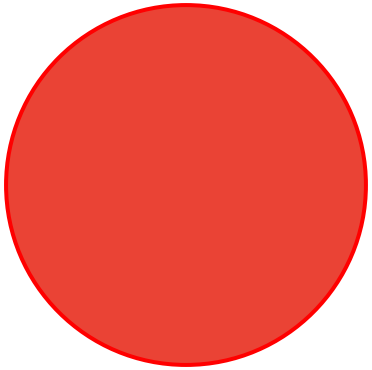
The movement in this space is not slowing down

MCP specification changes and new capabilities
Elicitation, authentication, and much more.

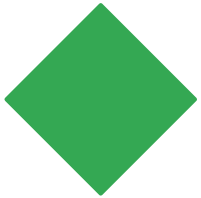
New model capabilities
Models continue to improve and enlarge their supported context window sizes.

User expectations and patterns
The entire industry is figuring out how best to work with AI tooling and MCP servers.





What's next for DevTools & MCP?

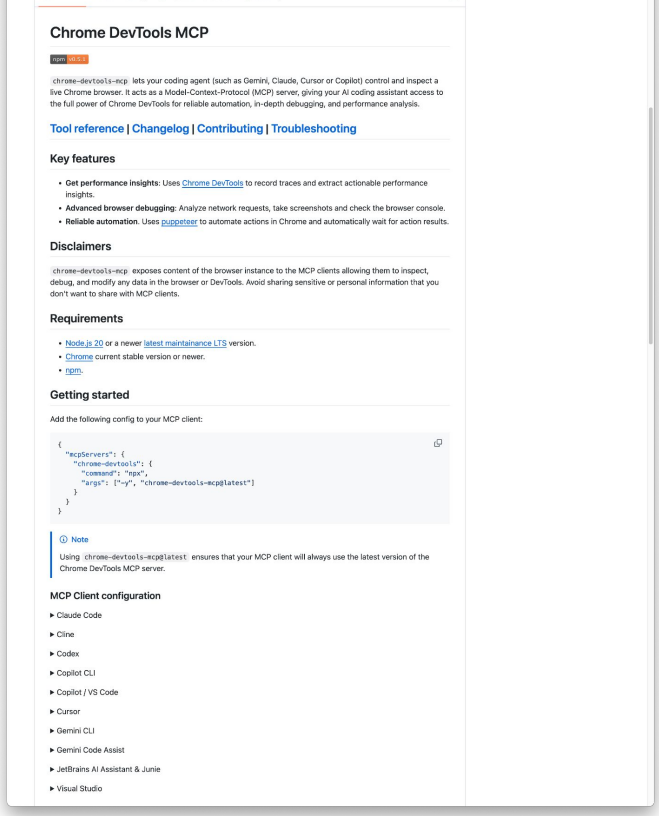


Please give us your feedback!

We want to continue evolving our MCP server to provide the best debugging experience for you and **your AI agents.**

Your feedback is extremely valuable and helps us prioritise features and fixes.

Please feel free to raise issues and tell us what's missing.



https://github.com/ChromeDevTools/chrome-devtools-mcp

Thank you

https://github.com/ChromeDevTools/chrome-devtools-mcp

jackfr@google.com