

WEB DAY
2026

Equip Your AI Agent with Chrome DevTools MCP for Smarter Debugging

Gilberto Cocchi

Staff Customer Engineer
Google



Do you use AI Coding
Agents for Web
development?

What about MCP?

Skills?

AI Coding Agents can test
your site? Right?

Dev: prompts AI Coding agent to do X

Agent: returns code for X for your website

Dev: Opens browser to verify whether it worked or not.
If not... Sending a new prompt

Agent: returns modified code

Dev: Refresh browser page and checks again whether it worked this time

Coding Blindfolded

Agents can't see real-time code results in a browser.

Creates back-and-forth churn.

Creates user frustration.

Costs tokens.

Introducing Chrome DevTools MCP server

Connects your AI agent directly to a live Chrome browser.

Gives your AI access to the power of Chrome DevTools.

github.com/ChromeDevTools/chrome-devtools-mcp



Ask the agent. Trust the runtime.

Key Features

- **Performance:** record traces and extract actionable insights
- **Advanced browser debugging:** Analyze network requests, take screenshots, console messages
- **Reliable automation:** integrates Puppeteer to automate actions



Recently released

- Experimental chrome-devtools CLI
- **Skills** support to onboard mcp server
- **Lighthouse** Audits
- New `-slim` mode for token savings
- **CrUX** data into perf trace summaries

github.com/ChromeDevTools/chrome-devtools-mcp



📁 * Browser Action

```
> /clear  
└ (no content)
```

```
> █
```



```
-- INSERT -- ▶▶ bypass permissions on (shift+tab to cycle)
```

User Registration - TechCorp x +

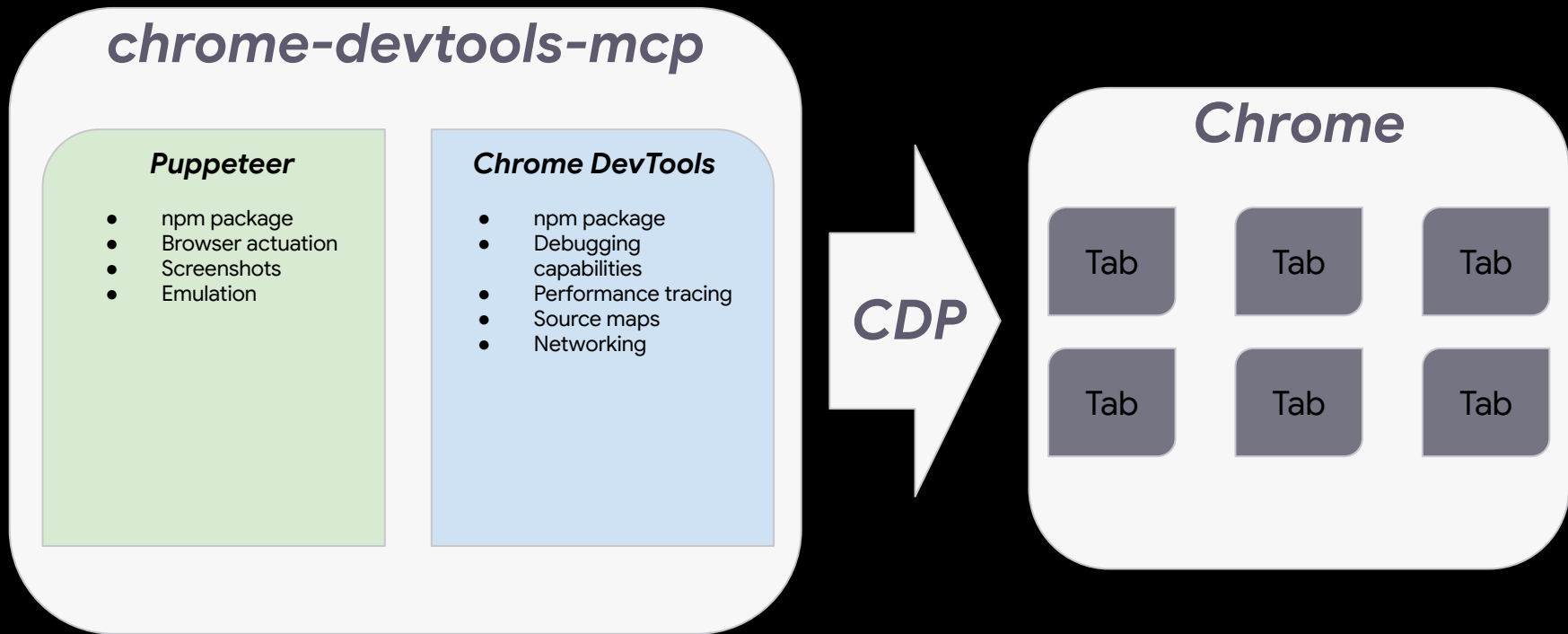
localhost:3000

Create Your Account

Join TechCorp Solutions today and get access to our premium services.

First Name *	Last Name *
<input type="text" value="aaaa"/>	<input type="text" value="bbbb"/>
Email Address *	Phone Number
<input type="text" value="aaa@mail.com"/>	<input type="text" value="+1 (555) 123-4567"/>
Password *	Confirm Password *
<input type="password" value="....."/>	<input type="password" value="....."/>
Date of Birth	Country
<input type="text" value="dd . mm . yyyy"/> 	<input type="text" value="Select a country"/> 

How does the MCP server work?



Getting started

```
{  
  "mcpServers": {  
    "chrome-devtools": {  
      "command": "npx",  
      "args": ["-y", "chrome-devtools-mcp@latest"]  
    }  
  }  
}
```

github.com/ChromeDevTools/chrome-devtools-mcp



How to install on Gemini-CLI

```
# Either MCP only:
```

```
gemini mcp add chrome-devtools npx chrome-devtools-mcp@latest
```

```
# Or as a Gemini extension (MCP+Skills):
```

```
gemini extensions install --auto-update https://github.com/ChromeDevTools/chrome-devtools-mcp
```

How to install on Claude Code

```
claude mcp add chrome-devtools --scope user npx chrome-devtools-mcp@latest
```

github.com/ChromeDevTools/chrome-devtools-mcp



Gemini-CLI Browser Agent (experimental)

```
{  
  "agents": {  
    "overrides": {  
      "browser_agent": {  
        "enabled": true  
      }  
    }  
  }  
}
```

geminicli.com/docs/core/subagents/#browser-agent-experimental



```
"args": [  
  "chrome-devtools-mcp@latest",  
  "--channel=stable", // Change Chrome channel.  
  "--headless=true", // Whether to run in headless (no UI) mode.  
  "--isolated=true", // Temporary user-data-dir that is automatically cleaned up after the browser is  
closed.  
  "--auto-connect=true", // Automatically connects to a browser running in the user data directory.  
  "--browser-url=.:9222", // -u Connect to a running, debuggable Chrome instance.  
  "--ws-endpoint=...", // WebSocket endpoint to connect to a running Chrome instance  
  "--ws-headers=...", // Custom headers for WebSocket connection in JSON format  
  "--executable-path=...", // -e Path to custom Chrome executable.  
  "--user-data-dir=...", // Path to the user data directory for Chrome.  
  "--log-file=...", // Path to a file to write debug logs to. Set the env variable DEBUG to * to  
enable verbose logs.  
  "--viewport=...", // Initial viewport size for the Chrome instances started by the server.  
  "--proxy-server=...", // Proxy server configuration for Chrome when launching the browser.  
  "--accept-insecure-certs=...", // If enabled, ignores errors relative to self-signed and expired  
certificates.  
  "-chrome-arg=...", // Additional arguments for Chrome.  
  "--ignore-default-chrome-arg=...", // Explicitly disable default arguments for Chrome.  
  "--category-emulation=...", // Set to false to exclude tools related to emulation.  
  "--categoryPerformance=...", // Set to false to exclude tools related to performance.  
  "--categoryNetwork=...", // Set to false to exclude tools related to network.
```

DEMO



What can you (or better, the agent) do?

Input automation: `click`, `drag`, `fill`, `fill_form`,
`handle_dialog`, `hover`, `press_key`, `type_text`, `upload_file`

Navigation automation: `close_page`, `list_pages`, `navigate_page`,
`new_page`, `select_page`, `wait_for`

Network: `get_network_request`, `list_network_requests`

Network: `emulate`, `resize_page`

github.com/ChromeDevTools/chrome-devtools-mcp#tools



What can you (or better, the agent) do?

Performance: `performance_analyze_insight,`
`performance_start_trace,` `performance_stop_trace,`
`take_memory_snapshot`

Debugging: `evaluate_script,` `get_console_message,`
`lighthouse_audit,` `list_console_messages,`
`take_screenshot,` `take_snapshot`

github.com/ChromeDevTools/chrome-devtools-mcp#tools



How does an AI agent see the page?

Accessibility tree light

take_snapshot function returns the A11y tree of the page.

Includes unique identifier for controls.

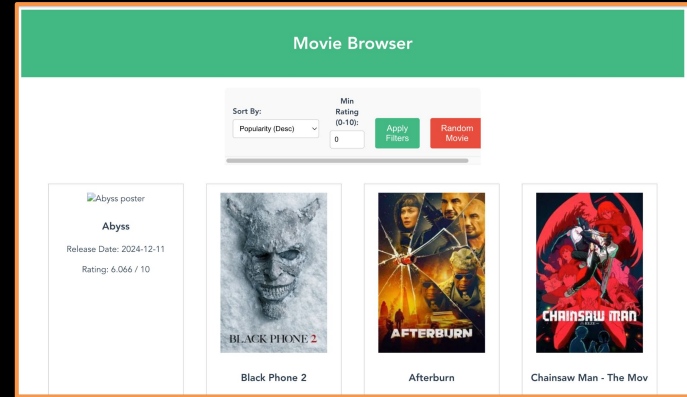
Includes selected element.

```
# take_snapshot response
## Page content
uid=4_0 RootWebArea "movie-browser" url="http://localhost:8080/"
  uid=4_1 banner
  uid=4_2 heading "Movie Browser" level="1"
  uid=4_3 main
    uid=4_4 StaticText "Sort By:"
    uid=4_5 combobox "Sort By:" haspopup="menu" value="Popularity (Desc)"
    uid=4_6 option "Popularity (Desc)" selectable selected value="Popularity (Desc)"
    uid=4_7 option "Popularity (Asc)" value="Popularity (Asc)"
    uid=4_8 option "Release Date (Desc)" value="Release Date (Desc)"
    uid=4_9 option "Release Date (Asc)" value="Release Date (Asc)"
    uid=4_10 option "Rating (Desc)" value="Rating (Desc)"
    uid=4_11 option "Rating (Asc)" value="Rating (Asc)"
    uid=4_12 StaticText "Min Rating (0-10):"
    uid=4_13 spinbutton "Min Rating (0-10):" value="0" valuemax="10" valuemin="0" valuetext="0"
```

Screenshots

take_screenshot returns a screenshot of the page or element.

Can be saved to a file instead too.



CyberAgent automating runtime error debugging

- Connected an AI agent (Claude) to Chrome DevTools via MCP
- agent autonomously navigated through stories, logs, identified errors, and implemented code fixes without human intervention
- Audited 32 components and 236 stories in approximately one hour allowing developers to focus on complex logic
- Process became the new standard for AI debugging at CyberAgent

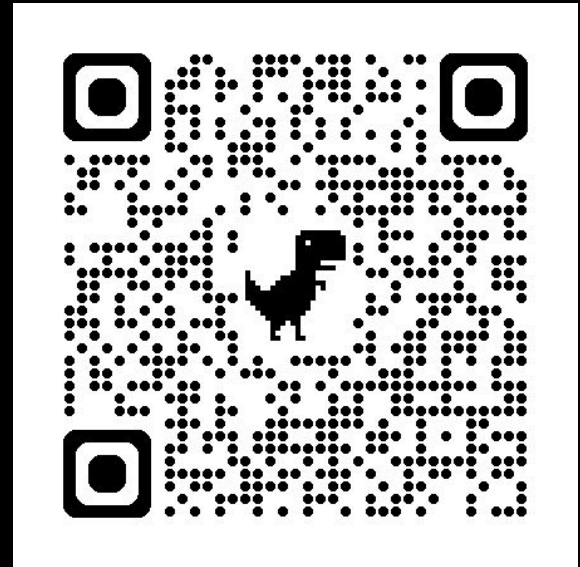
developer.chrome.com/blog/autofix-runtime-devtools-mcp



Want to try it out? Questions?
Feedback? Pull requests?

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

github.com/ChromeDevTools/chrome-devtools-mcp



WEB  DAY
2026

Thanks!