

How to Use Developer Tools (DevTools) in Atlas for Web Development

TechRounder PDF Edition

Live article:

<https://www.techrounder.com/how-to/how-to-use-developer-tools-devtools-in-atlas-for-web-development/>

By Vipin PG | Published November 14, 2025 | Updated January 11, 2026 | Format: Guide | 5 min read

Quick answer

If you're a web developer thinking about using ChatGPT's Atlas browser for your daily work, you'll want to know about some important limitations with its developer tools. Let's break down what works, what doesn't, and how you can make the most of what's available right now.

If you're a web developer thinking about using ChatGPT's Atlas browser for your daily work, you'll want to know about some important limitations with its developer tools. Let's break down what works, what doesn't, and how you can make the most of what's available right now.

What's the Current Situation with Atlas DevTools?

Atlas is an interesting browser because it combines AI capabilities with web browsing. However, when it comes to traditional web development work, it falls short in some pretty significant ways.

Developers across communities like Reddit have been vocal about these issues, and they're worth understanding before you dive in.

The Main Problems Developers Are Facing

1. You Can't Dock DevTools to the Side

This is the biggest headache for most developers. In Chrome, Firefox, Safari, and Edge, you've always been able to dock your DevTools as a sidebar while you work. It's become second nature to have your code on one side and your tools on the other.

In Atlas, that's not possible. DevTools only open in a separate window. One frustrated developer on r/OpenAI put it this way: "Having devtools open docked as a permanent sidebar has been standard in every major browser for many years. However, in Atlas, the devtools only appear in a separate window, which makes typical web development tasks feel quite cumbersome."

Here's what makes this particularly puzzling: Atlas is built on Chromium (the same foundation as Chrome), which means it should technically support sidebar docking. The fact that it doesn't suggests this was an intentional design choice rather than a technical limitation.

2. Device View Is Completely Missing

If you've ever tested how your website looks on different screen sizes, you've used Device View (sometimes called the Device Toolbar). It's that feature that lets you quickly switch between desktop, tablet, and mobile views.

In Atlas, this feature doesn't exist. Multiple developers have confirmed this absence, with one stating: "It's surprising that a solid set of developer tools for web developers doesn't come with a built-in device toolbar. Such a feature seems essential and should have been included from the outset."

This is a major problem because responsive design testing isn't optional anymore-it's fundamental to modern web development. Without Device View, you're flying blind when it comes to how your site performs across different devices.

3. No Way to Switch to Mobile View

Beyond just the missing Device View feature, developers report they "can't even switch to mobile view" at all in Atlas. This severely limits your ability to develop and test responsive applications properly.

Why Is This Happening?

The Short Answer: It appears to be a deliberate design decision, not a technical bug.

Since Atlas is built on Chromium, these features should be available automatically. One developer theorized: "Given that Atlas is built on Chromium, one would expect docking capabilities to be included by default. Could this limitation be a deliberate choice due to integrated ChatGPT sidebar?"

This makes sense when you think about it. The ChatGPT sidebar is Atlas's main selling point-it's what makes the browser unique. OpenAI may have decided that keeping this sidebar always available was more important than supporting traditional DevTools docking.

Workarounds You Can Try

While there's no official way to fix these limitations, here are some practical approaches that developers have found helpful:

Option 1: Use Multiple Monitors

If you have a dual-monitor setup, you can work around the docking limitation:

Downside: This requires extra hardware and desk space, which isn't feasible for everyone.

Option 2: Manual Window Management

On a single monitor, you can manually arrange windows side-by-side:

1. Resize your browser window to take up the left half of your screen
2. Resize the DevTools window to fill the right half
3. Use window management tools (like Magnet on macOS or FancyZones on Windows) to snap windows into position quickly

Downside: This requires manual setup every time and isn't as smooth as native docking.

Option 3: Master the Keyboard Shortcuts

Learn the DevTools keyboard shortcuts to minimize the annoyance of switching between windows:

Downside: This doesn't solve the fundamental problem, just makes it slightly less painful.

Is Atlas Ready for Professional Web Development?

The honest answer: Not yet.

The developer community has been pretty clear on this point. Here's what real developers are saying: "I was on the verge of switching browsers after two decades of using Chrome, but this issue has made me reconsider. I won't be using Atlas until this is resolved."

Another developer noted: "Every major browser for decades has supported sidebar docking as standard. The fact that Atlas doesn't include this is a significant oversight for a browser built on Chromium."

The consensus is that these aren't just minor inconveniences-they're fundamental workflow issues that make Atlas unsuitable as a primary development browser right now.

What's OpenAI Planning to Do About It?

The good news is that OpenAI knows about these issues and has plans to address them.

What we know:

What we don't know:

How Should You Use Atlas Right Now?

Here's a practical approach based on the current state of the browser:

Use Atlas as a Secondary Tool

Don't replace your main development browser yet. Instead, think of Atlas as a supplementary tool that brings AI capabilities to your workflow.

Keep using Chrome, Firefox, or Safari for:

Use Atlas for:

Maximize What Atlas Does Well

Even with its limitations, Atlas has some unique strengths:

The ChatGPT Sidebar can help you:

AI Features that can complement your development:

If You Have Multiple Monitors

Position Atlas and its DevTools on separate screens to minimize workflow disruption. This won't give you the same experience as proper sidebar docking, but it's the closest approximation available.

Stay Updated

Keep an eye on OpenAI's release notes and announcements. When they do add improved DevTools, you'll want to know about it quickly.

What You Can Still Do in Atlas DevTools

Despite the limitations, some basic DevTools functionality is still available:

You can still inspect HTML elements, check console output, and perform basic debugging. The functionality exists-it's just not as ergonomic or efficient as you're used to.

Alternative Approaches While Using Atlas

Here are some creative ways developers are working within Atlas's constraints:

The Bottom Line

Atlas is an innovative browser with genuinely useful AI features, but it's not ready to be your primary web development tool. The inability to dock DevTools and the missing Device View functionality are deal-breakers for professional development workflows.

Here's what you should do:

1. Keep your current browser (Chrome, Firefox, Safari, Edge) as your main development environment
2. Use Atlas as a companion tool for its AI capabilities and research features
3. Monitor updates from OpenAI about improved DevTools functionality
4. Revisit Atlas periodically to see if the limitations have been addressed

Once OpenAI implements proper DevTools support-which they've committed to doing-Atlas could become a compelling option that combines traditional development tools with AI assistance. Until then, it's best suited as a supplementary tool rather than your daily driver.

The good news is that OpenAI is actively developing Atlas and listening to developer feedback. These limitations won't last forever, but for now, it's important to understand what you're working with and plan accordingly.