

Recherche: Bester Browser für Web Development mit BricksBuilder

Kurzfassung / Empfehlung

Es gibt **keinen speziellen "Bricks-Builder-Browser"**, aber die Community und Performance-Tests zeigen ein klares Bild:

Einsatzzweck	Empfehlung
Bricks-Editor selbst (Backend, Drag&Drop, große Layouts)	Google Chrome oder Microsoft Edge – schlank eingerichtet, ohne Extensions, in eigenem Profil
Datenschutz + trotzdem performant	Brave (wird im offiziellen Bricks-Forum mehrfach als "läuft super mit Bricks" empfohlen)
Multi-Viewport / responsive Testing zusätzlich zum Editor	Polypane oder Sizzy – spezialisierte Dev-Browser
Standards-Check / Fallback bei Chrome-Bugs	Firefox

Der Grund: Bricks Builder ist im Backend eine **React/JS-Single-Page-App mit viel DOM-Manipulation** (Structure-Panel, Live-Preview-Iframe, Style-Panel-Reaktivität). Das bedeutet, die **JS-Engine- und Rendering-Performance des Browsers** ist entscheidend – und da liegt Chromium (Chrome, Edge, Brave, Vivaldi) klar vorne, weil V8 + Blink am optimiertesten für genau solche schweren SPA-Interfaces sind. Firefox ist standardkonform und stabil, aber bei sehr komplexen/verschachtelten Bricks-Seiten im Backend häufig spürbar langsamer laut Nutzerberichten im Forum.

Was das offizielle Bricks-Forum sagt

Im Thread ["Which browser is better to work in on windows"](#):

- Chrome frisst viel RAM
- Edge wurde als "läuft besser" beschrieben
- **Brave** wurde explizit gelobt: "Works great with Bricks and cares about your privacy"
- Firefox als Alternative für manche Workflows

Zusätzlich gibt es diverse Bug-Threads zu Performance-Problemen bei **komplexen Seiten mit vielen Elementen** ("Lag in the Builder when pages are complex", "Bricks slows browser to a freeze, uses insane RAM & CPU") – das ist aber primär ein **DOM-Struktur-Problem von Bricks selbst**, kein reines Browser-Problem.

⚠ Wichtiger bekannter Bug: Chrome ab Version 135

Seit **Chrome 135** gibt es ein neues natives Feature (`appearance: base-select` für `<select>`-Dropdowns, CSS-styleable Selects). Dieses Feature verursacht in Bricks + Advanced Themer nachweislich **erhebliche Lags im Editor**, weil die neue Rendering-Logik für Formularfelder mit der Bricks-Struktur kollidiert.

Fix/Workaround:

- In `chrome://flags` das entsprechende Select-Flag deaktivieren, oder
- Auf eine ältere Chrome-Version/Edge zurückgreifen, bis Bricks das gefixt hat
- Alternativ Brave nutzen (basiert auf Chromium, aber patcht solche experimentellen Blink-Features oft später/anders ein)

Spezialisierte "Dev-Browser" (deine Idee: super-schlank + Performance-fokussiert)

Es gibt tatsächlich genau das, was du suchst – allerdings nicht für den Bricks-Editor selbst, sondern für das **Testen/Entwickeln der fertigen Website**:

Polypane (polypane.app)

- Electron-App mit eingebautem Chromium
- Multi-Viewport-Preview (mehrere Breakpoints synchron), Accessibility-Checks, Design-Review-Tools
- Sehr gut für: Responsive-Testing deiner Bricks-Seiten, nicht für das Backend-Editing selbst

Sizzy (sizzy.co)

- "The Browser for Web Developers" – ähnliches Konzept wie Polypane
- Synchronisiertes Scrollen/Klicken über mehrere Device-Frames gleichzeitig
- Gut fürs schnelle visuelle Feintuning des Frontends

Wichtig zu wissen: Diese Tools sind selbst **Electron-Wrapper um Chromium** – sie sind also nicht "schlanker" oder performanter als reines Chrome/Chromium beim eigentlichen JS-Rendering. Sie bringen zusätzliche Features (Multi-View, Sync) mit, was für Layout-Testing extrem praktisch ist, aber für die reine Editor-Performance im Bricks-Backend selbst **keinen**

Geschwindigkeitsvorteil bringt – im Gegenteil, durch die zusätzlichen Panels/Features kann es sogar mehr RAM verbrauchen.

Praktische Tipps für maximale Performance im Bricks-Editor (unabhängig vom Browser)

1. **Eigenes, schlankes Browser-Profil** nur für WP-Admin/Bricks anlegen – keine Extensions (Ad-Blocker, Passwort-Manager, Dark-Reader etc.), diese greifen oft ins DOM ein und verlangsamen den Editor massiv
2. **Hardware-Beschleunigung** in den Browser-Einstellungen aktiviert lassen (GPU-Rasterization)
3. **DevTools geschlossen halten**, wenn nicht gebraucht – offene DevTools kosten deutlich Performance
4. **Cache regelmäßig leeren**, besonders nach Bricks-Updates
5. Lokale Entwicklungsumgebung mit SSD + Object-Cache (z. B. Local, LocalWP, oder Docker) nutzen – oft ist die eigentliche Bremse der Server/DB, nicht der Browser
6. Bei sehr komplexen Seiten: Seiten in kleinere Templates/Header-Footer-Includes aufsplitten, um die DOM-Komplexität zu reduzieren (das eigentliche Kernproblem laut Bricks-Bugtracker)

Fazit

Für die **reine Editor-Arbeit** in Bricks Builder ist ein **schlankes, extensionfreies Chrome-, Edge- oder Brave-Profil** aktuell die beste Wahl – kein Spezial-Browser schlägt hier die native Chromium-Performance. Wenn du zusätzlich **responsives Layout-Testing** super effizient machen willst, lohnt sich **Polypane** oder **Sizzy** als Ergänzung (nicht als Ersatz) neben deinem Haupt-Editor-Browser.

Revision #1

Created 2026-07-29 15:33:41 UTC by art10m

Updated 2026-07-29 15:34:30 UTC by art10m