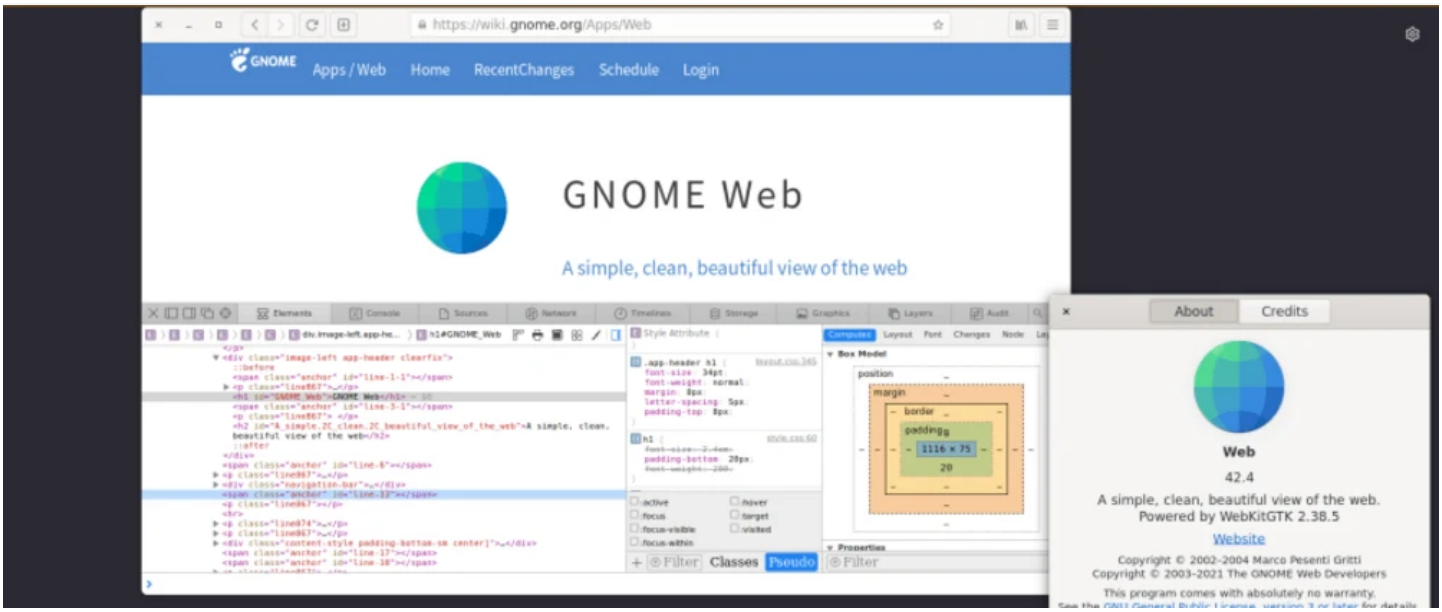


Ingo Steinke
Posted on Apr 13

💖 3 🦄 2 🤯 1 🙌 2 🔥 2

Fix Apple Bugs using a "Secret" Linux Browser

#webdev #testing #safari #beginners

Pragmatic Quality Assurance (4 Part Series)

1 Automating Tests using CodeceptJS and Testomat.io: First Steps

2 How to fix Frontend Tests as a lazy developer

3 Comparing Full Page Screenshots, Cross-Device

4 **Fix Apple Bugs using a "Secret" Linux Browser**

How to make sure that your site does not look broken in a specific browser? How to test cross-device without access to an extensive device lab or spending a lot of money for BrowserStack?

Cross-Device Testing

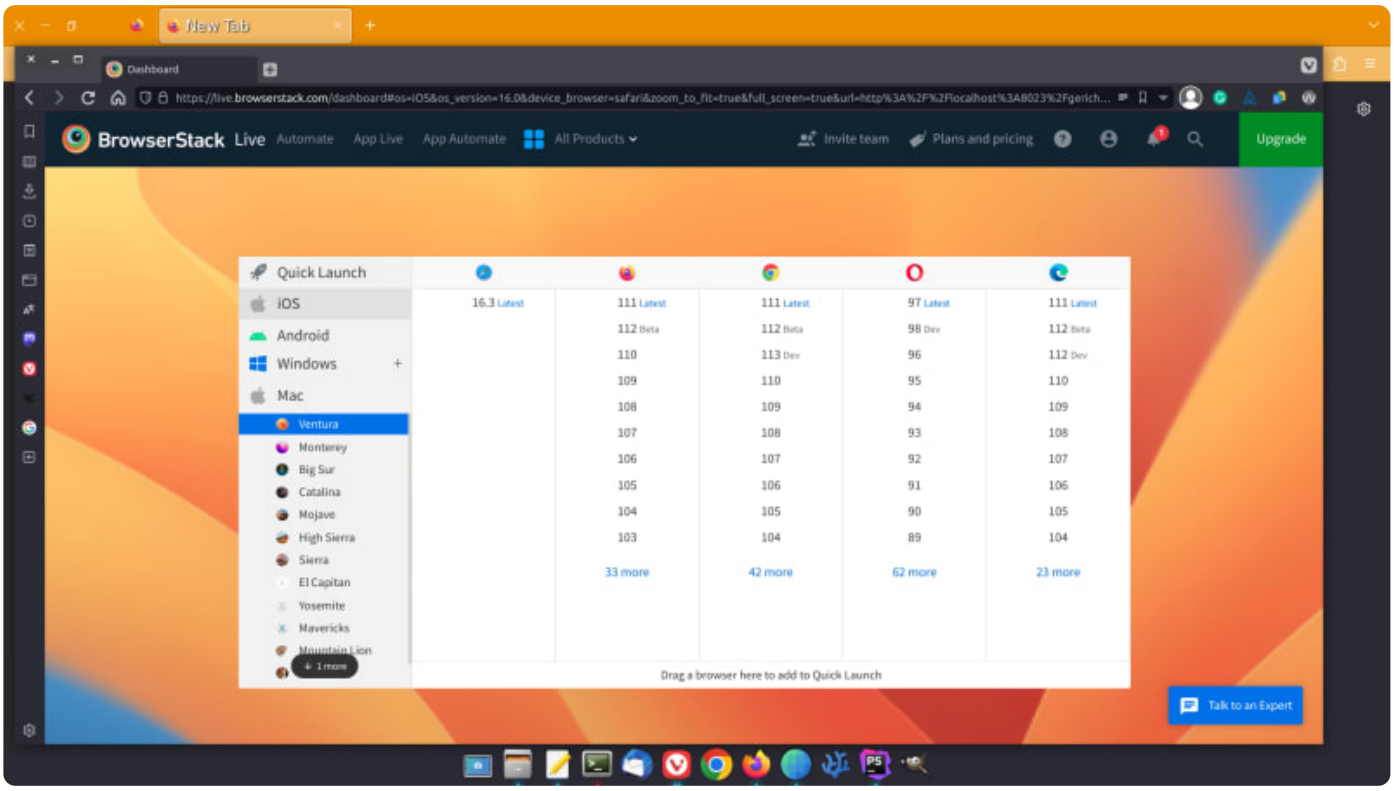
Well, if you want to earn money as a professional web dev, your you should spend money for professional tools - like BrowserStack. Currently it seems to be the only extensive device lab remotely available, and it even supports localhost servers and developer tools. My recent post about Testing a local WordPress Instance on BrowserStack mostly applies to any other website as well.

Testing a local WordPress Instance on BrowserStack

Ingo Steinke • Aug 3 '22

#webdev #testing #wordpress #devops

If you're on a tight budget, they offer a freelancer plan, including 100 (?) h per month - which can become less than you think, so you have to click quickly and always stop your session to return to the extensive device menu.



But you don't have to test every little feature on every posisble device! Read and study common problems so you can avoid, and more easily find and debug them without wasting your time double-checking features that have long been compatible. See caniuse.com. Ask your project manager which browsers need to be supported, and what's their definition of "supported". That might range from pixel-perfect to somehow usable without looking completely broken.

Focus on Known Bugs, Known Browsers, and Known Screen Sizes

Apart from a reasonable responsive web design approach, it might also help to consider what your designer, project manager, or customer might assume to be a "typical laptop screen width" like 1440 (CSS) pixel width on a 15 inch MacBook, and other MacBooks variing from 1366 to 1536. Likewise, there are different "typical" iPhone resolutions. Asking for a customer's / project manager's screenshot at an early stage of the project might give you a hint which device they are testing on, so you can do likewise.

You can set up automated testing, compare screenshots, and run audits like Lighthouse and axe to ensure that your site meets a common accessibility and web performance standard.

Don't only focus on a single browser and a preferred screen size, but don't waste your time and concentration trying to check every possible aspect all of the time either.

Edge Cases: Testing Microsoft Browsers on Linux

Use a virtual machine. Windows and Linux should be no problem to emulate in VirtualBox especially on a PC, and I heard that you can even set up a Hackintosh instance if you don't own a mac. Virtual machine used to be the last straw when adding those "we found out that, unlike the original specs we must support Internet Explorer 11", now Safari is the new IE. I know I must not say that, but I will unless they prove me otherwise.

Microsoft has done a great job getting rid of their old default browser and building he new EDGE so compatible to chrome that I rarely ever need to test or fix anything in EDGE at all. And it's compatible cross-platform so I don't need no virtual windows wasting resources and energy anymore.

Testing Apple Browsers as a Linux User

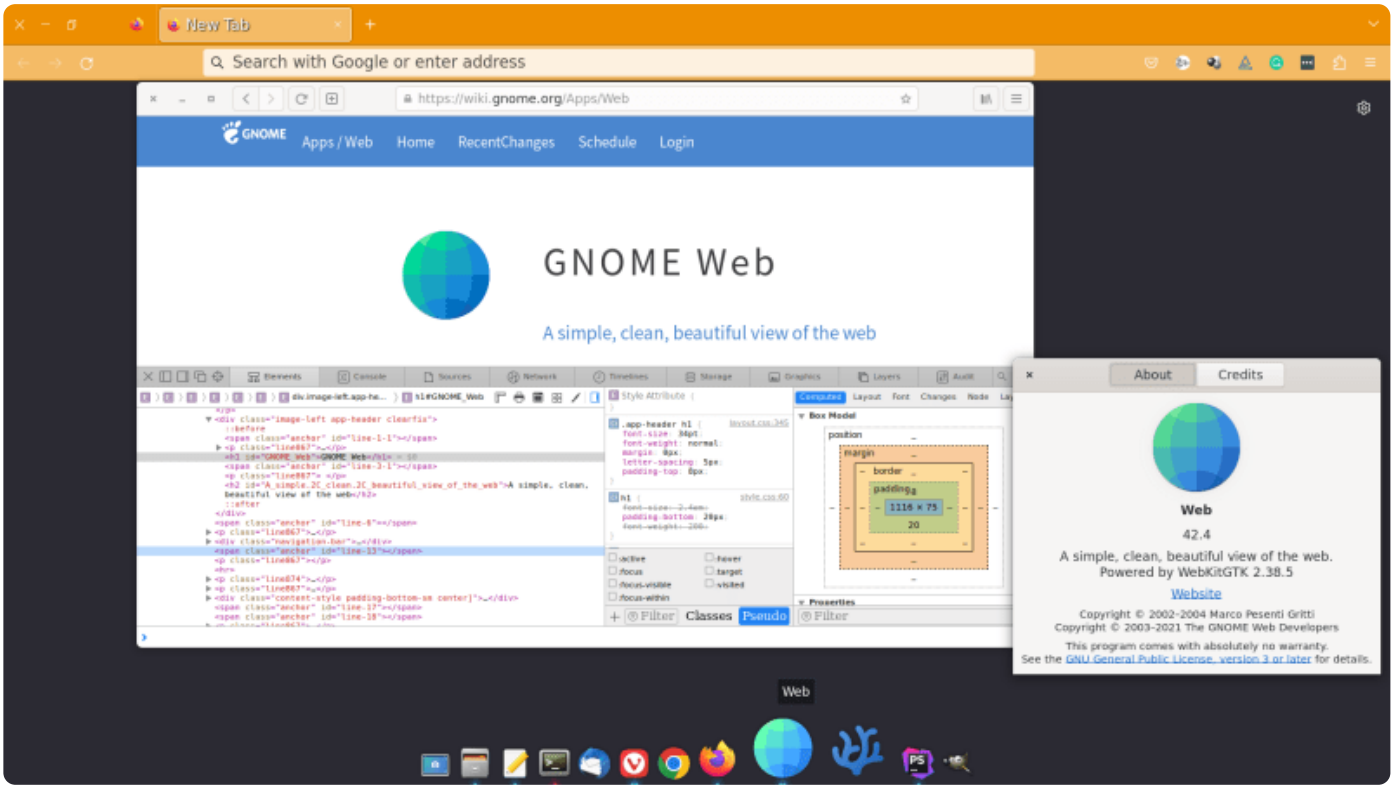
After the official end of Internet Explorer, we're mostly talking about Safari Bugs in the bugfixing phase of our projects. Despite the regular cheers about another ambitious feature first supported by Safari, you can't count on that Safari version to be shipped on your customers devices soon or ever. As an incentive to make people buy newer Apple devices although the old ones still work well, at least their hardware, Apple practices planned obsolescence so that cheered Safari version will never ever be available on an iPhone 8. And the latest original Chrome and Firefox won't either,

restricted by the "WebKit rule" preventing any alternative browser engine, so the alternative browsers on iPhone are, at the time of writing, nothing but UI skins with a Chrome or Firefox look and feel.

Keep those outdated machines or buy one for little money. Maybe you can get your hands on an old Macbook or a used iPhone. And Last but not least, and the original point to start this article: there are more browsers available than you might be aware.

Enter Gnome Web, an alternative WebKit Browser

Web (WebKitGTK) there is an alternative webkit browser that shares a lot of bugs, shortcomings, and peculiarities with Apple Safari. Simply called "Web" on my Ubuntu laptop, that "A simple, clean, beautiful view of the web.
Powered by WebKitGTK 2.38.5" has also been known as Gnome Web or Epiphany in alternative builds and former version. It's got a beautiful minimalist globe icon, and it can save you time and money fixing Safari bugs in a native application on your own machine without an actual Apple Safari.



Conclusion and Disclaimer

Disclaimer: don't get me wrong, Gnome Web is NOT and has never intended to be an Apple Safari replacement. It does NOT provide 100% compatible bugs and features. But it helped me to reproduce, understand, and fix several Safari bugs before finally testing them on an actual Apple device.

Some bugs and quirks are unique to the browser's limitations, like you might not see the intended fonts. I think it is a great idea to have an alternative testing browser that behaves differently from the default one. Now I've got at least three native desktop browsers: Vivaldi (Chromium), Firefox, and Gnome Web.

If a site looks good or broken in Gnome Web, that does not imply anything about its behavior in Apple Safari. But at least it helps us to develop our web apps in a way that we don't have to argue that it "works on my machine" when others are testing it in their browsers.

Pragmatic Quality Assurance (4 Part Series)

1 Automating Tests using CodeceptJS and Testomat.io: First Steps

2 How to fix Frontend Tests as a lazy developer

3 Comparing Full Page Screenshots, Cross-Device

4 **Fix Apple Bugs using a "Secret" Linux Browser**

Top comments (2)

Ben Sinclair • Apr 13

You have to bear in mind that the Browserstack tools, especially local testing, have their own exciting bugs as well, so while it's a reasonable place to start, it's not comprehensive.

Ingo Steinke • Apr 13

Thanks for the reminder, of course they have. And so does local testing in general as well, as some licensed web fonts and other third party components might not behave like expected on a local host.

In the end, nothing compares to real user experience on actual devices.

Code of Conduct • Report abuse