

Chrome Dev Summit: Mobile Summary



By Paul Kinlan

Paul is a Developer Advocate

The Chrome Dev Summit finished a couple of weeks ago, and here's the first in a series of reports from the event. There was a strong emphasis on Mobile and Cross-device development, so we'll kick off with that!

Best UX patterns for mobile web apps by Paul Kinlan

After an analysis of the mobile-friendliness of the top 1000 sites we found some problem areas: 53% still only provide a desktop-only experience, 82% of sites have issues with interactivity on a mobile device and 64% of sites have text that users will have issues reading.

Quick hits to dramatically improve your mobile web experience:

- Always define a viewport
- Fit content inside the viewport
- Keep font sized at a readable level
- Limit use of Web Fonts
- Size and space out tap targets appropriately
- Use the semantic types for input elements

PageSpeed Insights just launched a UX analysis for determining how mobile-friendly your site is. It will help you find common problems with your sites mobile UX. Try it out!

Slides: Best UX patterns for mobile web apps

Multi-device Accessibility by Alice Boxhall

Users will be accessing your sites and services from a multitude of devices with a wide range of different accessibility requirements. By using the correct semantic elements and correct ARIA roles you help give the browser and assistive technology a much improved understanding of your page.

Slides: Multi-device Accessibility.

Key ways to understand and address a11y issues

- Ensure you have a good keyboard-only user experience
- Express the semantics of your interface with correct element choice and ARIA
- Use ChromeVox [↗](#) on desktop and TalkBack on Android to test.
- Try the Accessibility Developer Tools Chrome extension
- A more diverse audience is getting online, which further amplifies the need of making your sites accessible

Build Mobile Apps using the Chrome WebView by Matt Guant

We all know the problems that developers have had in the past building for WebView: Limited HTML5 features, no debugging tools, no build tools. With the introduction of a Chromium powered WebView in Android 4.4 (KitKat) developers now have a huge range of new tools at their disposal to build great native apps using the WebView.

The WebView supports full remote debugging with the same tools you use for Chrome. You can also take your trusted web development workflow with Grunt and integrate that into

your native stack tooling via Gradle. Further merging worlds, there's a clever trick to use the Chrome DevTools to test your native code from JavaScript.

Slides: Build Mobile Apps using the Chrome WebView

Effective WebView development takeaways

- It's not the new features that are important, its the tooling that you can now use to speed up your workflow
- Don't try to emulate the native UI. But make sure to remove some of the tells that it is Web Content.
- Use native implementations of features when appropriate. i.e, use the DownloadManager rather than XHR for large files.

Optimizing your Workflow for a Cross-device world by Matt Gaunt

If we have to develop for Desktop, Mobile, Tablet, wearables and other form factors, how can you optimise your workflow to make your life less stressful? There's a solid multi-device approach for quick iteration with LiveReload, Grunt, Yeoman, and the newly-unveiled Mini Mobile Device Lab. Lastly, if you don't have the physical hardware you want to test, some providers make it available through the cloud.

Slides: Optimizing your Workflow for a Cross-device world

Key points

- The number of devices that we are going to have to cater for is only going to increase
- Getting your workflow with the right with [Grunt](#) [↗](#) and [Yeoman](#)
- Simplify cross browser and cross device testing with [Mini Mobile Device Lab](#)
- Be smart with your emulation choice using Chrome DevTools Emulation, Stock Emulators, Cloud Based Emulators like [Saucelabs](#) [↗](#), [Browserstack](#) and [Device Anywhere](#) and third party emulator [Genymotion](#)
- Mobile testing means more than just testing on your wifi connection, use a proxy to simulate slower network speeds

Network connectivity: optional by Jake Archibald

We learnt many things from this talk: Jake doesn't wear shoes when presenting; [Business Kinlan](#) has a new book coming out soon; Offline is being taken seriously by browser vendors and you will soon have the tools in your hands that help you build great experiences that work well when you are offline.

[ServiceWorker](#) will give us the flexibility that we need to build compelling offline first experiences with ease and not suffer the pains inflicted by AppCache. You can even [experiment with the API using a Polyfill](#).

[Slides: Network connectivity: optional](#)

ServiceWorker to the rescue

- In the next generation of progressive enhancement, we treat the network as a potential enhancement
- ServiceWorker gives you full, scriptable, debuggable control over network requests

- If you have an offline experience, don't wait for the network to fail before you show it, as this can take ages

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 2, 2018.