Add-ons for Firefox Android

Everything You Need to Know!
Today’s Speakers

Giorgio Natili  
Director of Engineering

Shane Caraveo  
Engineering Manager

Simeon Vincent  
Senior Developer Relations Engineer
1. Why Android, Why Now?
2. What Is Changing
3. Release Goals
4. User Experience
5. Toward the Release
6. Next Steps and Call to Actions
7. Questions and Answers
Why Android, Why Now

A journey in the motivations of this initiative
The importance of an open ecosystem

Mozilla operates in the open and delivers open source products.

The open-source nature of Firefox has allowed for a large community of developers to contribute to its development, resulting in a highly customizable and secure browser.
Mozilla operating values

Mozilla operating values are the foundations of our work.

Mozilla is committed to an internet that includes all the peoples of the earth, promotes civil discourse, human dignity, and individual expression.
Individual expression and customization

**Mozilla** is committed to an internet that promotes individual expression.

Individual expression and **customization** are correlated in that customization allows individuals to **express themselves** in unique ways.
We meet our users where they are and need us

Mozilla is committed to an internet that includes all the peoples of the earth.

Meeting users where they are and need Firefox helps to ensure that the internet community can access the products they want wherever they are.
Mobile growth

Android and Firefox are globally available software.

As of September 2023, Android is the most used mobile operating system worldwide, with a market share of 69.76% and Firefox users are growing on mobile by ~20%.
What Is Changing

How Firefox Android changed to extend add-ons support
Multi-processes architecture

Firefox increased code execution isolation as part of the reaction to Spectre and Meltdown.

We are extending support for the extension child process to Firefox for Android. This change enables a multi-process architecture for extensions that fits into the Android processes lifecycle.
Android lifecycle and event pages

The Android lifecycle and Firefox event pages are both important concepts in software architecture but are not directly related.

The Android lifecycle is a series of states an app goes through as it is launched, used, and closed. Event pages allow developers to write background scripts responding to events such as clicks, page loads, and tab switches.

They are the perfect fit for the Android process lifecycle.
Low resources and user control

Android manages processes by prioritizing foreground over background execution.

Android kills background processes that consume excessive resources. We have implemented an experience that lets the user decide to shutdown add-ons.
Release Goals

Enabling our open ecosystem on mobile devices
Add-ons on mobile devices unlock new capabilities

Add-ons are like apps that run in the browser and have a wider distribution market.

Opening up what Firefox already supports on desktop enables developers’ creativity to delight Firefox users.
Opening Android add-ons to all developers

Previously, Android users had access to a curated collection of recommended extensions.

Our goal is to enable any developer to build and make their add-ons available on Android.
Support & enable the add-ons community

We’re not just making it possible to build extensions, we’re building tools and resources to help developers along the way.

- Android testing recommendations [mzl.la/webext-android-ux](mzl.la/webext-android-ux)
- Browser API compatibility table [mzl.la/webext-compa](mzl.la/webext-compat)
Opening up Android is just the start

Our **focus** is on opening the extension ecosystem on Android **as it exists today**. That means:

- Manifest version 2
- Event pages
- Some APIs are missing
Opening up Android is just the start

There’s more to come

Next, we will expand and improve the WebExtensions platform on Android. This includes:

- Manifest version 3
- Expand API support

We need developer feedback to prioritize what we focus on next.
User Experience

How to find, install, and manage add-ons on Android devices
Recommended content

Expanded Add-ons catalog

The recommended extensions are available in Firefox Android release since August 2020.

On the same page, the user will get access to all the add-ons that developers marked as compatible with Android.
Searching for new content

Search the entire catalog

All the extensions marked compatible with Android are visible in the search bar.

Results are organized by category, relevance, and name as in the desktop experience.
Installation and permissions

Seamless UX

Installing add-ons on Firefox Android is a smooth experience consistent with the current desktop one.

The user is prompted to confirm required permissions during installation and the extension can request optional permissions at runtime.
Firefox Android add-ons manager

Manage add-ons

All the installed add-ons are managed via the add-ons manager.

The manager includes notifications for blocked add-ons and settings to enable or disable add-ons in private mode.
Toward the Release

A challenging timeline that preserve user experience and safety
A commitment to our users and developers

The Web Extensions and Android Engineering teams are focused on the critical features needed to expand add-ons on Android.

While doing this, the team built processes to support developers and community to streamline the development and adoption of solutions that personalize the mobile browsing experience.
A seamless experience is paramount

Stability, security, and performance are key components for an extensible and personalizable mobile browsing experience.

The new architecture has been getting tested and monitored for 3 months+ in Nightly, Beta and Release, starting with Firefox 118.

Manual QA and automated functional and performance testing combined with developer outreach is in progress.
Enabling more and more add-ons

Starting with Firefox 119 Nightly and Beta users can install over 75 add-ons.

On top of monitoring background and foreground crashes that exceed a threshold, Nightly and Beta users are contributing to crowd testing the available add-ons.
Developer outreach and support

Developers are receiving feedback from QA testing and getting support from the add-ons community and evangelism team.

The outstanding response from the dev community is critical to enable additional content in Nightly and Beta. Developers are getting 1:1 email support, office hours, webinars, and tutorials to make their extensions ready for Android (what about t-shirts?).
Next Steps and Call to Actions

What to expect and how to contribute
Get involved!

For developers, we have put together several resources to assist in preparing their desktop extensions for Android - those resources are available in one of our recent blog posts.

We invite end users to participate in our add-on community testing effort! Details for participating may be found here and all are welcome.
Questions and Answers
Development resources

Forum
discourse.mozilla.org

Documentation
extensionworkshop.com

Office hours
extensionworkshop.com/community
Deep-dive resources

- Non-persistent background scripts
  MDN

- Converting to event pages
  Tutorial

- Firefox process model
  Documentation

- GeckoView thread topography
  Documentation
Thank you!