
Best practices for developing for Chromebook tablets

The goal of this guide is to share some best practices for developing Android and web apps for Chromebook tablets. With the help of educators and developers around the globe, we've learned some tips and tricks along the way and want to share those with you and your team. Chromebook tablets come in many different shapes and sizes and run all sorts of software—from traditional websites to extensions, progressive web apps, and Android apps.

In many ways, Chromebook tablets operate the same as a traditional Chromebook. The operating system is largely the same. They're managed the same and run the same apps. Educators tell us that the smaller and lighter form of the tablet and no built-in mouse or keyboard changes how they and students use the device. For example:

- Many schools use tablets in early elementary, special education, and media classes where entering text, typing, and testing are less prevalent.
- Tablets are more conducive to handwriting notes, reading, filming, photography, and touchscreen apps than traditional laptops.
- Some schools prefer to use Android apps with tablets (because they're designed for touchscreens) instead of using websites (usually designed for a mouse).

This guide aims to help you adapt your apps to support these differences.

Best practice 1: Google Drive integration

One of the things that has made Chromebooks and Chromebook tablets so successful in schools over the years is that students can sign in to any device and access their files anywhere, anytime—such is the nature of the web!

Android apps tend to work a little differently because they run natively (offline). Files are often stored inside the app (as opposed to on a server or in Drive). In a shared-device setting, this means that students have to track down the device that they previously worked on to access their data.

To address this challenge, we encourage developers to store student projects in Google Drive so that data synchronizes across apps and devices. This [open-source project](#) demonstrates the functionality. The project:

- Automatically writes student data to a visible folder in Drive using the [REST API](#)
- Syncs local data with the Drive folder when the app launches

- Identifies all files by [ID#](#) versus file name in case the student edits the folder or contents

Adopting this practice ensures that students and schools can control their data and that students can pick up any Chromebook tablet and access their work.

If you want to store student projects on your own server, we encourage offering a way for students to sign in with their Google Account so that young students don't have to remember and enter app-specific usernames and passwords.

To save files on the native Chrome filesystem from an Android app, you can use the [Storage Access Framework](#).

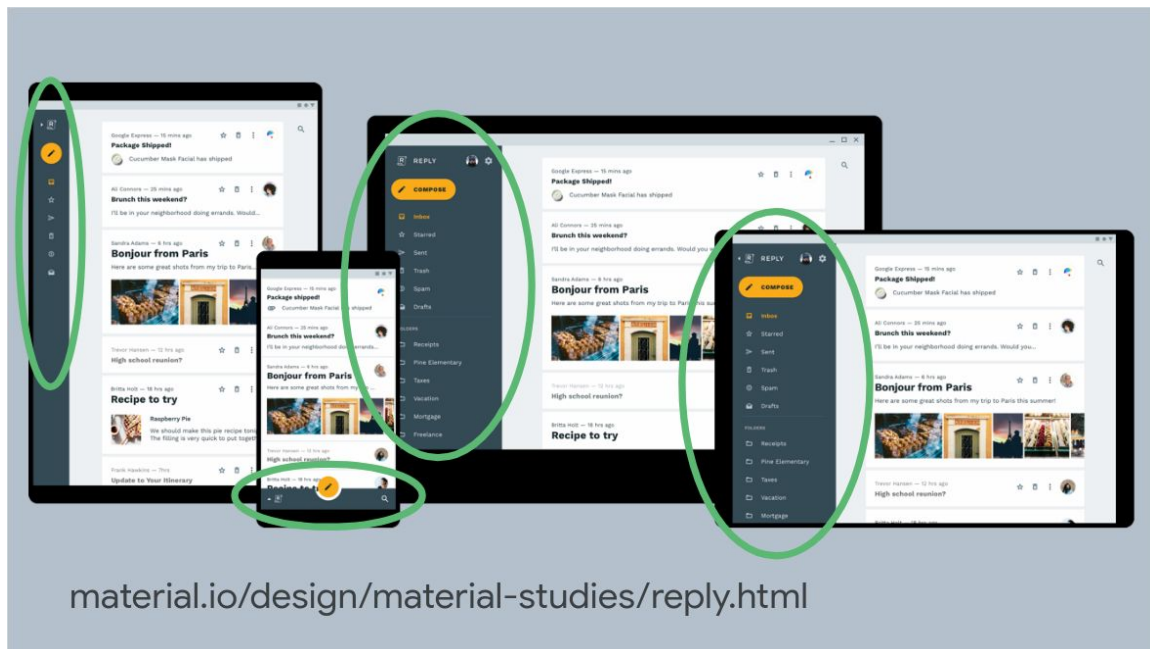
An example of an app that has utilized these functionalities well is [Explain Everything](#). The app gives users the option to save files to their local storage and Google Drive.

Best practice 2: Optimize UI for Android apps

To the surprise of many developers, modern Chromebook tablets run Android apps right alongside websites, progressive web apps, and Chrome extensions. The apps can be downloaded from the Google Play store and deliver graphic-rich games, augmented-reality experiences, and interactive learning activities designed for touchscreens.

For the most part, your Android app should just “run” on a Chromebook tablet, and we recommend some best practices for ensuring a great user experience on a tablet.

- **Make your app resizable:** Apps should smoothly adapt to the device they're on. Certain apps can benefit from running in [fullscreen/immersive](#) mode. For example, learning games for kids often run better in fullscreen mode to prevent distractions and accidental navigation away from the app.
- **Adapt navigation patterns:** Depending on the screen size and orientation, different navigation patterns (bottom bar, side navigation) are more appropriate. For examples, go to the [Reply app](#) and the [Chrome OS navigation demo](#).



- **Include keyboard and mouse support:** Many tablets have detachable keyboards that are used with bluetooth-enabled keyboards and touchpads. Include expected keyboard and mouse actions (keys and tab navigation). For details, review [Handle keyboard actions](#).
- **Review these other resources:**
 - a. [Android apps on Chromebook tablets](#)
 - b. [Optimizing Android apps for Chromebook tablets](#)
 - c. [Stylus support for Chromebook tablets](#)
 - d. [Responsive grid layout in material design](#)

Best practice 3: Optimize UI for web apps

If you have a web app that you want to optimize for Chromebook tablets, we have a number of suggestions for making sure it's touch-friendly.

- **Maintain minimum touch targets:** Any on-screen element that someone can click, touch, or otherwise interact with should be large enough for reliable interaction. Make sure these elements have a width and height of at least 48 dp, as described in the [Material Design Accessibility guidelines](#).
- **Support touch events:** Most websites are designed for a mouse. Tablets don't have a mouse. Some web apps, such as Adobe Spark, implement these principles and include a full-screen button that helps users focus on creativity. For details, review [Add Touch to Your Site](#).

- **Support a stylus:** If your app includes drawing tools, [this MDN article](#) describes the API that you need for incorporating stylus support into the app.
- **Upgrade to a progressive web app:** Give your web app native-like capabilities, reliability, and installability by turning it into a [progressive web app](#) like [Book Creator](#).

Best practice 4: Build with accessibility in mind

We encourage developers and publishers to design and build products and apps with accessibility in mind. To build accessible apps, review these [developer resources](#).

Best practice 5: Help schools find your app

One of the questions we often hear from educators is, “Where can I find classroom tools for Chromebook tablets?” We heard you loud and clear and launched [Chromebook App Hub](#) to showcase all the great educational tools created by developers around the world that work on Chromebooks and Chromebook tablets.

To learn more about the Chromebook App Hub complete [this form](#).

Marketing Development Funds opportunity

We want to help developers who focus on a strong Chromebook tablet experience. The Google for Education team provides Marketing Development Funds to up to 30 app developers to help accelerate development and promote education apps. The funds are reserved for developers who have:

- Optimized their app’s user interface (UI) for tablets
- Ideally implemented Google Drive integration
- Joined the Google Cloud Partner Advantage program under the Build engagement model

You can use the funds for activities, such as case studies, events to promote a touch-optimized app, or a digital marketing campaign to promote apps on tablets.

In the first part of 2020, we will work directly with you on your application for Marketing Development Funds. If you’re interested, email gfe-technology-partnerships@google.com.