

# CHROME EXTENSION KIT



# Table of Contents

<b>INTRODUCTION .....</b>	<b>3</b>
ABOUT CHROMEEXTENSIONKIT .....	3
WHAT'S INCLUDED IN THE KIT? .....	3
SHOWCASE.....	4
OFFICIAL BLOG .....	4
FUTURE KIT UPDATES .....	5
WHAT IF I NEED HELP OR SOMETHING ISN'T WORKING? .....	5
<b>CHOOSING A STARTER TEMPLATE.....</b>	<b>7</b>
BASIC STARTER TEMPLATES .....	7
REACT STARTER TEMPLATES .....	8
<b>EXAMPLE EXTENSIONS .....</b>	<b>12</b>
DAY COUNTER .....	12
PASSWORD GENERATOR .....	12
WELCOME – NEW TAB .....	13
SEARCH TEXT .....	13
<b>GETTING PRODUCTION READY .....</b>	<b>15</b>
PREPARING THE EXTENSION CODE.....	15
PREPARING NECESSARY IMAGES.....	16
UPDATING YOUR EXTENSION .....	17
<b>PUBLISHING YOUR EXTENSION .....</b>	<b>20</b>
<b>GAINING TRACTION .....</b>	<b>27</b>
SETTING YOUR EXTENSION UP FOR SUCCESS.....	27
SHARING YOUR EXTENSION .....	30
LISTENING TO YOUR USERS AND STATS .....	32
<b>UNDERSTANDING EXTENSION STATS .....</b>	<b>36</b>

# Introduction

# Introduction

## About ChromeExtensionKit

Chrome Extensions are powerful tools that can take what's possible in Chrome to a whole new level and with lots of remaining opportunity in the extension space, the question often gets asked, where and how do I start building one?

I built my first Chrome extension roughly 4 years ago and asked many of the same questions. After some trial and error, I was able to grow my first extension to over 3,000 active users and have since applied what I learned to future extension projects where similar results were achieved. This kit includes a series of starter templates that I created for my own projects after getting tired of repeating the same setup process for each new project. The idea is simple: figure out what type of Chrome extension you want (new tab, popup, etc.) and grab the appropriate starter template. No config, no hassle, just jump in and get started as fast as possible.

Not only can you pick the perfect extension starting point template out of the numerous options, you can also easily grab bits from other templates and merge them together or take a look at any of the fully functional example extensions to see how things are done or to get some inspiration.

## What's included in the kit?

The kit includes a number of resources that allow you to get started with Chrome Extension development as quickly as possible. These resources include:

- 8x Basic HTML / CSS / JS starter templates which include the following starting points:
  - Content scripts
  - Context Menu
  - Messaging
  - New tab override
  - Browser action for popup and full page
  - Browser action for popup only
  - Browser action for full page only

- Chrome API usage (storage)
- 9x React starter templates (with scripts for easy publishing preparation and development) which include the following the following starting points:
  - Content Scripts
  - Data fetching
  - Browser action for popup and full page
  - Browser action for popup only
  - Browser action for popup only (with TypeScript)
  - Browser action for full page only
  - Chrome API usage (storage)
  - New tab override
  - Fully loaded React app base extension (React, React Router, and Tailwind)
- This PDF that includes the following:
  - Explanation on how to use the kit, the starter templates, and the example extensions
  - Explanation on how to prepare your extension for publishing
  - Web store publish guide
  - Tips on growing your extension and gaining traction
  - Brief introduction to extension stats and metrics as well as how to use them

## Showcase

I'm currently working on compiling a list of extensions that were started with one of the templates in this kit. The showcase will be featured on the main website at:

<https://ChromeExtensionKit.com/showcase/>.

If you're interested in being added to the showcase, reach out to me with your extension at [Ryan@ChromeExtensionKit.com](mailto:Ryan@ChromeExtensionKit.com) and I'd love to include it!

## Official Blog

If you're interested in keeping up with everything related to the kit as well as Chrome Extensions as a whole, feel free to check out the official blog at <https://ChromeExtensionKit.com/blog>. The blog will also include detailed explanations of major additions to the kit.

## Future Kit Updates

When purchasing the kit, if you selected "*ChromeExtensionKit may send me product updates and offers via email. It is possible to opt-out at any time.*", you will receive an email when changes are published to the kit. If you did not select this option, you can still find a list of updates, what they were, and when they happened on the site at <https://ChromeExtensionKit.com/changelog>.

If you originally did not select the option, but want to be emailed *strictly* about kit updates, feel free to reach out at [Ryan@ChromeExtensionKit.com](mailto:Ryan@ChromeExtensionKit.com) and I'll make sure you're included.

It's also important to note that your original download link included in your purchase email will *always* include the most up-to-date version of the kit.

## What if I need help or something isn't working?

If at any point you get stuck or find a bug, I would be glad to offer my assistance. Feel free to first checkout the FAQs at <https://ChromeExtensionKit.com/> and if your answer isn't there, reach out to [Ryan@ChromeExtensionKit.com](mailto:Ryan@ChromeExtensionKit.com) and I will get back to you as soon as possible.

# Choosing a Starter Template

# Choosing a Starter Template

## Basic Starter Templates

The *Basic Starter Templates* included in this kit utilize basic HTML, CSS, and JS only. They are best suited for those who don't want or need to utilize any specialized UI frameworks and are comfortable working with the basics. These starter templates can be found in the `/basic` folder and include the following:

### Content Scripts Starter

- Includes a basic starting point for an extension that uses content scripts, which are files that run in the context of certain web pages you define
- Can be found in `/basic/basic-content-scripts`

### Context Menu Starter

- Includes a basic starting point for an extension that adds to the user's context menu (i.e. right click menu)
- Can be found in `/basic/basic-context-menu`

### Browser Action Starter (popup and full page)

- Includes a basic starting point for an extension that triggers a popup when clicking on the extension's icon in the browser bar
- Moreover, inside the toggled popup, a full page can also be triggered with a button click (i.e. to open a settings page, for example)
- Can be found in `/basic/basic-popup-full-page`

### Browser Action Starter (popup page only)

- Includes a basic starting point for an extension that triggers a popup when clicking on the extension's icon in the browser bar
- Can be found in `/basic/basic-popup-page`

### Browser Action Starter (full page only)

- Includes a basic starting point for an extension that triggers a separate full page when clicking on the extension's icon in the browser bar
- Can be found in `/basic/basic-full-page`



### Messaging Starter

- Includes a basic starting point for an extension that utilizes messaging between a background script and a content script
- Can be found in `/basic/basic-messaging`

### New Tab Override Starter

- Includes a basic starting point for an extension that overrides the user's new tab with the provided HTML page
- The page can also be toggled by clicking on the extension's icon in the browser bar as well
- Can be found in `/basic/basic-new-tab`

### Chrome Storage (Chrome API) Starter

- Includes a basic starting point for an extension that makes use of the Chrome API and more specifically, sync storage (for persisting data across multiple sessions and browsers)
- Can be found in `/basic/basic-sync-storage`

In general, the process for setting up and working with any of the above templates locally is the following:

1. Navigate to `chrome://extensions` in your browser
2. Turn Developer Mode on (typically in the top right of the page)
3. Click "Load Unpacked" in the top left
4. Select the folder for the starter you want to use
5. You should now be running and ready to start making changes! Each change will require the extension to be reloaded on the `chrome://extensions` page

If you are unsure which of the above starter templates are most appropriate for your use case, you can easily test each one out as they include some basic functionality to get the idea across. Each folder includes a *README* with all the relevant setup and details for the given template.

## React Starter Templates

The *React Starter Templates* included in this kit are ideal for those who want to develop a Chrome extension using React (and optionally, TypeScript). Using React in the extension ecosystem is a little different than a typical web app, however these starter templates provide you with all the scaffolding and setup you need to avoid having to configure it, such as scripts for a development server that auto-refreshes the Chrome extension on change and a script that builds and prepares

your extension for publishing. These starter templates can be found in the `/react` folder and include the following:

#### **Content Scripts**

- Includes a basic starting point for an extension that renders a React app inside a given website via content scripts
- Can be found in `/react/react-content-scripts`

#### **Data Fetching**

- Includes a basic starting point for an extension that utilizes data fetching and network calls on a separate page
- Can be found in `/react/react-data-fetching`

#### **Browser Action Starter (popup and full page)**

- Includes a basic starting point for an extension that triggers a popup when clicking on the extension's icon in the browser bar
- Moreover, inside the toggled popup, a full page can also be triggered with a button click (i.e. to open a settings page, for example)
- Can be found in `/react/react-popup-full-page`

#### **Browser Action Starter (popup page only)**

- Includes a basic starting point for an extension that triggers a popup when clicking on the extension's icon in the browser bar
- Can be found in `/react/react-popup-page`

#### **Browser Action Starter (popup page only with TypeScript)**

- Includes a basic starting point for an extension that triggers a popup when clicking on the extension's icon in the browser bar
- Includes TypeScript support out of the box
- Can be found in `/react/react-typescript-popup-page`

#### **Browser Action Starter (full page only)**

- Includes a basic starting point for an extension that triggers a full page when clicking on the extension's icon in the browser bar
- Can be found in `/react/react-full-page`

#### **New Tab Override Starter**

- Includes a basic starting point for an extension that overrides the user's new tab with the provided React app

- The app can also be toggled by clicking on the extension's icon in the browser bar as well
- Can be found in `/react/react-new-tab`

### Chrome Storage (Chrome API) Starter

- Includes a basic starting point for an extension that makes use of the Chrome API and more specifically, sync storage (for persisting data across multiple sessions and browsers)
- Can be found in `/react/react-sync-storage`

### Fully Loaded React Setup

- Includes a more complex React app starting point with React, React Router, and Tailwind CSS already configured and setup. This starter is meant to be more fully loaded and opinionated than the other starters.
- Can be found in `/react/react-fully-loaded`

In general, the process for setting up and working with any of the above templates locally is the following:

1. Navigate to the folder with the starter you're interested and run `npm install` to install all the required dependencies
2. Next, run `npm run dev` to start the local development server (this creates new builds after every code change and outputs them to `/build`)
3. After, Navigate to `chrome://extensions` in your browser
4. Turn Developer Mode on (typically in the top right of the page)
5. Click "Load Unpacked" in the top left
6. Select the `/build` folder that should have been created within the current templates folder. This folder will be regenerated with every change you make as the Chrome extensions require static files to run, not React components
7. You should now be running and ready to start making changes!

If you are unsure which of the above starter templates are most appropriate for your use case, you can easily test each one out as they include some basic functionality to get the idea across. Each folder includes a *README* with all the relevant setup and details for the given template.

# Example Extensions

# Example Extensions

All example extensions are included under the MIT License.

## Day Counter

*Day Counter* is a Chrome extension that appears on each new tab where you can create counters that tracks days until or since life events, such as Birthdays. Counters can be re-ordered by a simple drag-and-drop feature and can be optionally set to automatically repeat weekly, biweekly, monthly, or yearly.

There are also a number of settings in order to customize your experience, such as changing the counter size, the color scheme of the page, filtering visible counters to only future or past counters, filtering counters based on date range (i.e. only showing counters that are a week away), and a privacy filter that blurs the counters on load and on inactivity so others cannot read them when you aren't there.

This extension is a real extension that has been live in the app store for a number of years and currently has over 3,000 active users. I included the source code for this extension to show what a fully functional and feature complete extension looks like that also supports a good number of users in production. The live version can be found in the store: <https://bit.ly/3fliSUJ>

This extension can be found in `/example/day-counter`.

## Password Generator

*Password Generator* is a React-based Chrome extension that displays a small popup when clicked and you can set settings and generate a password easily once opened. You can set your desired password length and whether or not it should include numbers and symbols. After generating the password, you simply click the field and it will auto-select, making it easy to copy.

Every time you re-open the extension, a new password will be generated based on the last selected parameters and settings.

This extension was created to help demonstrate some common concepts you will use in a Chrome extension, such as saving user settings. It was also used to show the entire publish process (See **Publishing Your Extension** section).

This extension can be found in `/example/password-generator` and the dist version was fully generated via the build and assemble script found in all React starters.

## Welcome – New Tab

*Welcome – New Tab* is a simple Chrome extension that overrides the new tab functionality and shows a random background image on each load. The extension also requests your name so it can welcome you by name on each load. Currently, the images are preloaded within the extension, but could easily be augmented by connecting it to the Unsplash API for free and using a much larger image set as a result.

This extension has not been published in the store and was solely created for informational purposes.

This extension can be found in `/example/welcome-new-tab`.

## Search Text

*Search Text* is a simple Chrome extension that lets a user highlight text on any webpage and then simply right click it and click “Search for X”. After that, a new tab will be created searching for that text on their preferred search engine. This extension utilizes a number of different Chrome APIs.

This extension has not been published in the store and was solely created for informational purposes.

This extension can be found in `/example/search-text`

# Getting Production Ready

# Getting Production Ready

## Preparing the extension code

The first step in preparing your extension to be published is preparing the code itself. This can mean different things depending on how your extension is built (i.e. basic HTML / CSS / JS versus using React, etc.), but the general idea is the same: you will need a finalized zip file containing all your *required* (more on this below) and static extension code plus the manifest file and icons.

### Manifest File

The manifest file is required for all Chrome extensions. It is recommended to add as many options as you see appropriate, however the minimum fields that must exist in all manifest files are:

- **"name"**: The name of your extension which displays in the Chrome web store and in the browser for users
- **"version"**: The current version of your extension. Each new version of your extension will require an incremented version, so it is suggested to start with something small and use semantic versioning, such as *1.0.0*
- **"manifest\_version"**: The version of the manifest to use. All starters and example extensions have been upgraded to use Manifest V3.

For more information on the manifest file: <https://bit.ly/2PntnfW>.

### Required Code

Your extension's zip file should only contain the code that is actually required for your extension to run properly and no other development code or files. This code should also just be static and non-obfuscated code (HTML, CSS, and JS).

For example, if you are developing an extension with SCSS, your zip file should only contain the compiled CSS, not the SCSS as well.

Another example is a SPA extension, such as one built with React. Your zip file should only contain the production build folder (which has the static assets) and the manifest file / icons and nothing else (such as components or other development files). If you use one of the React starter



templates included in this kit, there is a script built in that will create the production build and zip the required contents for you.

## Preparing necessary images

Now that you have your code zipped and ready to go, the next step is preparing necessary images. There are 3 types of images that relate to extensions: App Icons, Promotional Images, and Screenshots.

### App Icons

The app icon should technically be prepared when you are getting your extension production ready as at least one icon size must be included in the zip file for your extension. The Web Store recommends the following (at the very least):

- 128 by 128 pixels app icon with only 96 by 96 pixels being the icon itself and the remaining 16px on each side providing a buffer
- The icon should be in PNG format
- The icon should look good on both light and dark backgrounds (as this depends on how the user has Chrome configured)


For more information on app icon images: <https://bit.ly/2Xm9aLC>.

### Promotional Images

The promotional image is the initial image a potential user will see when browsing the store and looking for extensions to install. For example, the image to the left here is the promotional image for Day Counter appearing in the search results:

## Extensions

[More extensions](#)



### Day Counter - New Tab Page

Offered by: Ryan Fitzgerald

Track days until or since life events like Birthdays easily.

★★★★★ 47 Productivity

[Add to Chrome](#)

The Web Store requires at least one 440 by 280 pixels promotional image (seen above in the screenshot). With that said, you can also choose to provide a large promotional image (920 by 680 pixel) and a marquee promotional image (1400 by 560 pixel). It is recommended to provide the large and marquee images as well because they make your extension eligible to be featured in the Chrome Web Store which can be a huge traffic increase to your extension's page if you get chosen.

For more information on promotional images: <https://bit.ly/3gDeC4J>.

## Screenshots

The last type of images are the screenshots. Each extension must provide at least 1 screenshot, but it's recommended to provide the maximum amount, which is 5. Screenshots are used to show potential users what your extension is all about and they appear on your extension's store page. The recommended size for screenshots is 1280 by 800 pixels, but you can also use 640 by 400 pixels. Screenshots will ultimately be downsized automatically to 640 by 400 pixels for display, so try to ensure they look good when shown at a slightly smaller resolution.

For more information on screenshots: <https://bit.ly/3k4vF1w>.

# Updating your extension

I included a brief section on updates because I think it's important to mention a couple points:

## Updating Version Numbers

As mentioned, you will need to update the version number of your extension in the manifest file with each new upload to the Chrome Web Store. Starting with a small version number and using semantic versioning will save you from having your version number creep up to huge numbers over time (no one wants version 300 of their extension out there). For those unfamiliar with semantic versioning, it works by providing version numbers in the form of `[MAJOR].[MINOR].[PATCH]`, such as 1.0.0.

According to [semver.org](https://semver.org), increment the:

- MAJOR version when you make incompatible API changes,
- MINOR version when you add functionality in a backwards compatible manner, and
- PATCH version when you make backwards compatible bug fixes.

## Updating Screenshots

Although most likely an obvious one, you should always keep your screenshots up to date as your extension changes and progresses over time. The last thing you want is negative reviews popping up because a user downloaded an extension based on the screenshots and the extension looks nothing like them any more since they haven't been updated in some time. You are also likely improving and continuing to add to your extension over time, so take those up-to-date screenshots and show it off!

## Backwards Compatibility

One portion of the update cycle that often gets overlooked is backwards compatibility. One example of this is with the usage of Chrome storage due to their key / value nature. You need to ensure that if you make a change that alters how data is stored or retrieved, you still provide some backwards compatibility mechanism (whether it's wiping it on first load for old users or just loading multiple formats for a while). If this is neglected, all previous users who have downloaded your extension and had data be saved will run into issues when they update it with your latest published version.

An example scenario is if you're storing user settings in Chrome storage. If you initially use the key `settings` to store the value under and in a later version decide to change to `userSettings`, all users who have previously downloaded your extension will run into an issue if you don't provide a backwards compatibility mechanism as they will have data stored under `settings` in their browser but your extension after the update is expecting it under `userSettings`.

# Publishing Your Extension

# Publishing Your Extension

You've built your extension and prepared it to be published, you're almost there! The publishing process on the Chrome Web Store is fairly straightforward and can be summarized in the following steps:

1. Preparing your extension's zip file
2. Creating a developer account
3. Uploading your extension files
4. Adding your assets and other relevant information
5. Submitting your extension to be published

## 1. Preparing your extension's zip file

The first step in the publishing process is zipping up all your required extension files. This has been covered briefly in the **Getting Production Ready** section above. To summarize:

- Basic HTML / JS / CSS extension: manually create the zip by moving all the relevant files to a folder and compressing it (remember to remove anything that's not necessary, such as SCSS files if you added them or maybe a gulpfile, etc.)
- React extension: simply run `npm run assemble` which will create a production build and copy all the other required files (such as the manifest) automatically to a folder and compress everything into `extension.zip`.

## 2. Creating a developer account


In order to publish anything to the Chrome Web Store, you'll need a Google account (i.e. Gmail) that's been setup as a developer account as well. In order to do so, after logging into your Google account, navigate to the Chrome Web Store Developer Dashboard (<https://chrome.google.com/webstore/devconsole/register>).

If your account is not currently setup as a developer account, you'll see the **Register as a Chrome Web Store developer** message that looks like:

## Register as a Chrome Web Store developer

Welcome to the Chrome Web Store Developer Dashboard. Once registered, you'll be able to distribute, monetise and manage extensions and themes for the Google Chrome Browser. [Learn more](#)


Get started:



### Accept developer agreement and privacy policies

- [Google Chrome Web Store developer agreement](#)
- [Google Privacy Policy](#)
- [Chrome Web Store Developer Programme Policies](#)

☐ I have read and agree to the developer agreement and policies



### Pay the US \$5.00 registration fee

A one-time registration fee is required in order to register your account.

[Pay registration fee](#)

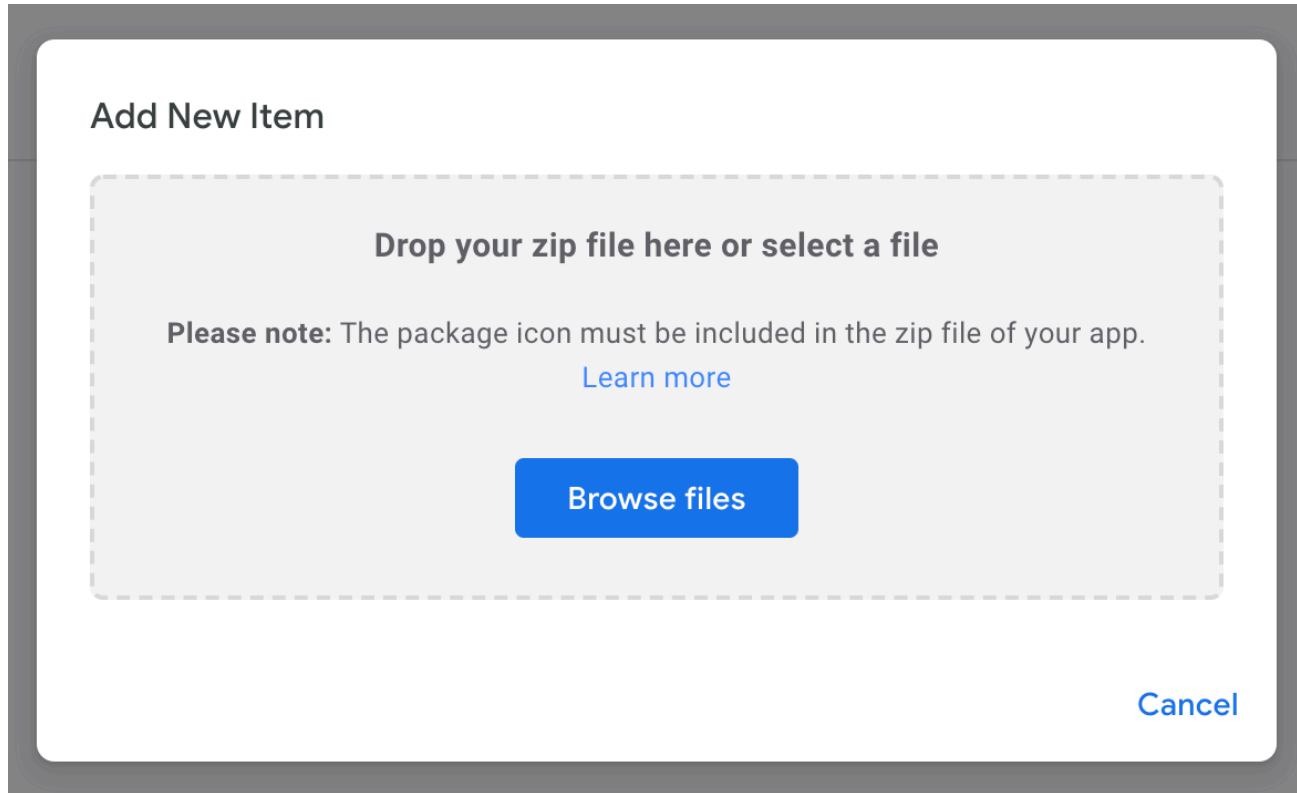
As you can see, you'll need to both **accept the developer agreement and policies** as well as pay a one-time fee of **US \$5.00** (this is used to reduce the amount of spam accounts appearing in the store). Once complete, you'll be brought to the Developer dashboard.

## 3. Uploading your extension files

To begin the process of uploading your extension, click the "New Item" button.



Next, select or drag your zip file created back in step 1 into the popup:



If there are any errors with how your zip file has been created, you will get notified at this stage. Common errors include things such as obfuscated code, forgetting your manifest file, forgetting app icons, etc.

#### 4. Adding your assets and other relevant information

If everything is successful, you should then be brought to a store listing draft page:

Package

Store listing

Privacy

Pricing & distribution

🔑 Password Generator

Status: Draft ID: ffbhpmommmmmnbnjdgcfdcpnllkeahh

[Why can't I publish?](#) [Save draft](#) [Submit for review](#)

Store listing

Product details

Title

For all languages

Password Generator

Title from Package

Summary

For all languages

Generate a password in seconds.

Summary from Package

Description \*

For all languages

Description\*

0 / 16000

Focus on explaining what the item does and why users should install it

Category \*

For all languages

Select a category

Language \*

[Learn more](#)

Select a language

Specifying your item's language will help users find it. If you support multiple languages, then you should internationalize your item.

Graphic assets

Screenshot ordering on this dashboard is not yet available. For now please use the [old dashboard](#) to order screenshots.

You'll notice some fields, such as the title and summary, can't be edited and that's because certain information is pulled from the manifest file directly. At this point, you can add the basic product details into each of the fields listed.

After the product details are complete, you'll see the Graphic assets section, which is where you can add promo videos, screenshots, and promo images:

Graphic assets

Screenshot ordering on this dashboard is not yet available. For now please use the [old dashboard](#) to order screenshots.

Global assets

Global promo video

Enter a YouTube video URL

Screenshots \*

1280x800 or 640x400  
JPEG or 24-bit PNG (no alpha)  
At least one is required.  
Up to a maximum of 5.

+

Drop Image Here

Small promo tile

440x280 Canvas  
JPEG or 24-bit PNG (no alpha)

+

Drop Image Here

Large promo tile

920x680 Canvas  
JPEG or 24-bit PNG (no alpha)

+

Drop Image Here

Marquee promo tile

1400x560 Canvas  
JPEG or 24-bit PNG (no alpha)

+

Drop Image Here



After adding all your assets and product information, you'll see a final "Additional fields" where you can choose to add some additional information such as a website URL, support URL, Google analytics ID, and more.

You'll also notice there are 3 other tabs to the left other than the current "Store Listing" tab. They are:

- Package Tab: displays basic info about the package you uploaded (version number, permissions you requested, etc.)
- Privacy Tab: allows you to add additional information relating to privacy within your extension
- Pricing & Distribution Tab: allows you to set information such as the visibility of your extension (private, unlisted, public) and distribution (all regions by default or only select some regions)

Throughout the edit process you can also save your progress as a draft in the top right corner and return later.

## 5. Submitting your extension to be published

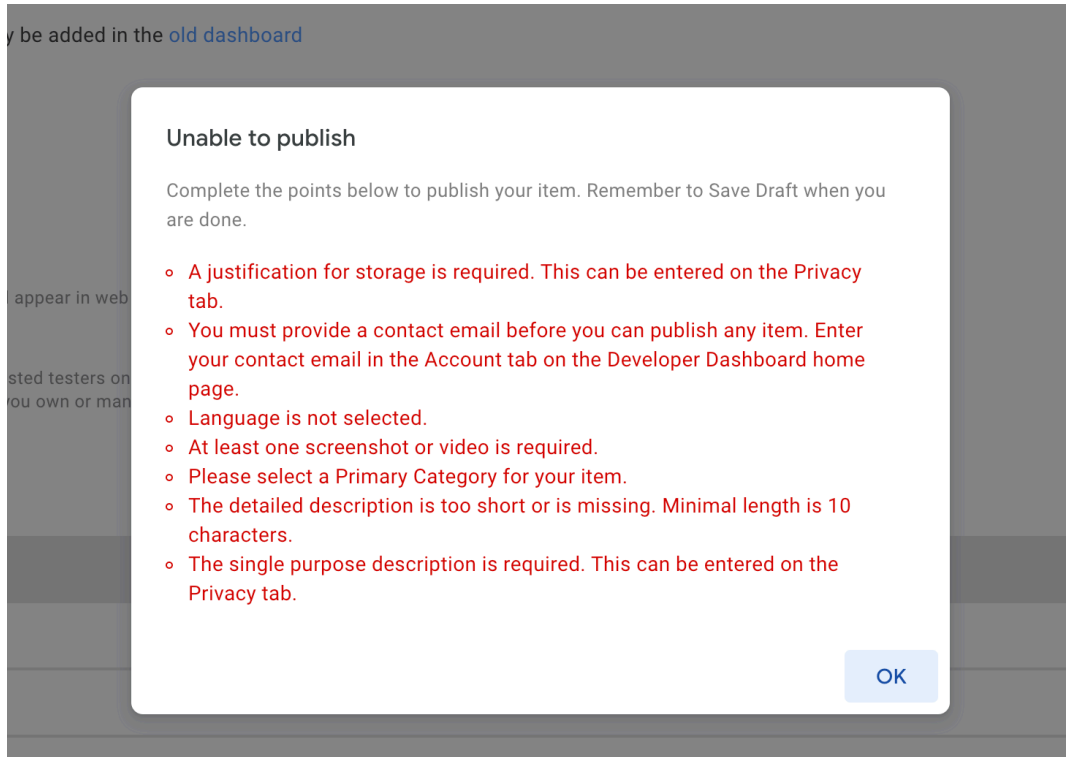
When you've added all the information, you'll find the "Submit for review" button in the top right corner:

 [Why can't I publish?](#)

Save draft

Submit for review

You may notice the button is greyed out, like in the screenshot above. If this is the case for you, it simply means you have some additional information that is required or there is an error at some point in the information you entered. To see what needs to be completed, click on "Why can't I publish?":



Once all the issues have been tackled, you'll see the "Submit for review" button is no longer greyed out and can be clicked on. When ready, click the button and you'll see a final popup asking you to confirm you're ready to submit for review. You can also select the option to publish automatically after it has passed review.

That's it, you're done! Congrats! In terms of review time, this can vary on a number of factors. If there are no issues with your submission, you should get a notification after some time and your extension will be live in the store.

# Gaining Traction

# Gaining Traction

Gaining traction and building a user base on a Chrome extension isn't an easy task and it will often take time to reach a critical mass where user growth is consistent and increasing over time, however there are a number of ways you can increase your odds of success and expedite the process slightly. There are 3 primary methods that will be discussed here: setting your extension up for success, sharing your extension, and listening to your users and stats.

## Setting your extension up for success

Gaining traction and building a user base for your extension starts with the extension itself and the extension web store page. Thinking about both of these components will help you maximize the amount of organic growth your extension gets.

On the extension end, is your extension complete? Does it look polished? Does it work well? Or is it just something you slapped together in a couple hours? There's nothing wrong with quickly throwing together an extension, however if it doesn't look polished or has a number of bugs, you can't expect it to perform quite as well among potential users. On the flip side, if you have a polished product that works well and solves a problem, you're likely to experience some degree of growth (even if just word of mouth) through that alone.

Aside from the extension itself, another way to set yourself up for success is to optimize your extension's web store page. There are a number of common optimizations that can be made:

### Title

The title of your extension is often overlooked but it provides an easy optimization that can have a big impact on your ranking in the store. One of the easiest improvements that can be made is using a couple keywords in your extension title itself. For example, I originally had Day Counter simply titled "Day Counter", however when I made the switch to renaming the extension "Day Counter – New Tab Page", I saw some noticeable improvement in my rankings.

In general, my advice for the title is to keep it short and try to summarize your idea in a couple keywords within the title. The pattern I like to follow for all my extensions is:

<Extension Name> - <Brief Extension Summary>

If you take a look at some of the top extensions in the store, you'll see it's a fairly common pattern. For example, here are a couple extension names taken from browsing the store quickly (all have over 100,000 users):

- Planyway - Calendar and Timeline for Trello
- Forest - stay focused, be present
- Screencastify - Screen Video Recorder
- Huntr - Job Search Tracker

You get the idea. Not only does it increase search ranks, it also tells your potential users right away what your extension is all about.

### Summary and Description

Much like the title, it's important to put some thought into your extension summary (defined in the manifest) and description (defined in the web store). The extension summary should, as the name suggests, summarize your extension in a couple of words with the help of keywords. The summary typically appears when a user hovers over your extension in the search result, so properly explaining your extension here will have a positive impact. The extension description should go into some additional detail; however, you don't want to provide a wall of text for a potential user to read. Aim for roughly a paragraph or two that goes into some more detail and explains some key features about your extension and try to use some keywords here as well.

In general, my advice for summary and description sections is to be succinct and try to utilize keywords as much as *naturally* possible in them. Although some users will likely not read the description or summary at all (and will just look at the screenshots), using keywords in both of these sections will help your ranking within the store, so it's still a must.

### Category Selection

Category selection is an easy one, but it should also have some thought put into it. The most important thing is to make sure you choose the best category that represents your extension as users will often search by category. If you find yourself between two different categories, both of which are equally valid, my suggestion would be to try the category with less competition.

### Screenshots

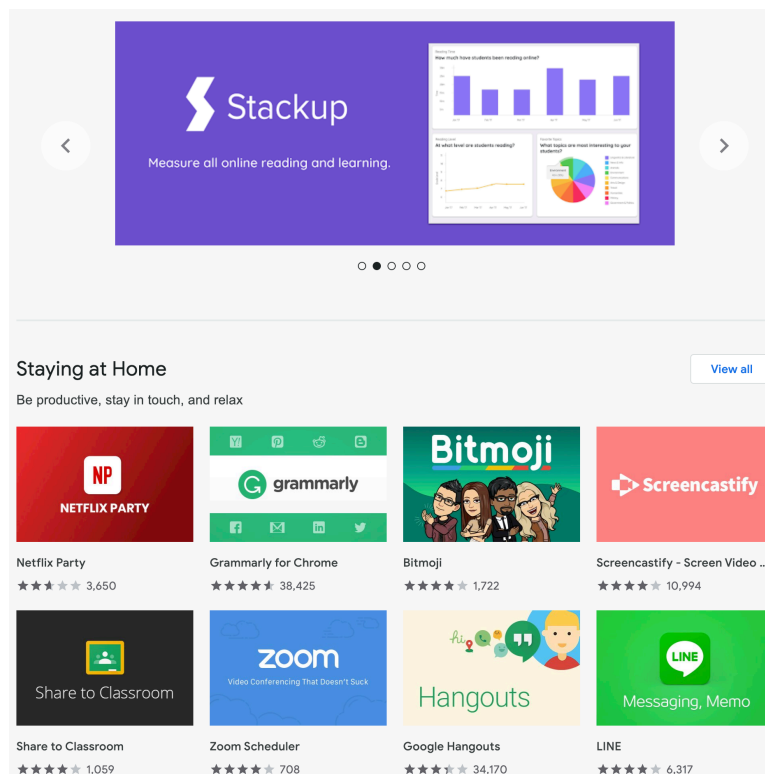
Take the time to really put together some proper screenshots. Not only should they look nice and professional, they should also accurately depict what your extension is all about (and ideally some of

its key features). Many potential users will often quickly scan over screenshots and not even read the description, so making sure you have awesome screenshots is a critical stepping stone for the success of your extension. I highly recommend you add as many screenshots as you can (5 is the current maximum).

Another suggestion I like to offer regarding extension screenshots is to take your screenshots and send them to someone you know without much context about your extension. If they can tell you generally what your extension is about just by looking at the screenshots, you likely have some solid screenshots.

## Promotional Images

Much like the screenshots, putting some time into producing quality promotional images is also important. Unlike screenshots, the promotional images typically don't show off your extension quite as much (with the exception of the marquee size) and are more a means of marketing and branding your extension. Let's take a look at an example of the web store:



You'll notice there are two types of the promotional images being used here: the marquee promotional image size (top carousel) and the small promotional image size (the bottom grid with all the extensions). Notice how the smaller sized promotional image is purely branding whereas the marquee is a mix of branding and a brief screenshot of the extension since there is some more space.

In general, my advice for promotional images is to see how others are doing it in the store and try to replicate something similar in terms of professionalism and branding. It's also important that you provide all 3 possible sizes of promotional images (small, large, and marquee) even though only small is required. The reason for this is if you don't provide a large and marquee size, you won't be eligible for potentially being featured in the web store (like Stackup in the example above). Being featured would mean a huge influx of traffic, so don't remove the possibility of it happening by not supplying those extra sizes.

## Sharing your extension

Now that you've set yourself up for success by optimizing your extension and its web store page, it's time to share it with the world! Getting the word out about your extension is an important step in helping get it off the ground. There are a number of methods you can use to share your extension with others.

### Online Communities

Many great online communities exist where you can potentially drive large amounts of traffic to your extension if done correctly. One of the biggest mistakes often made regarding sharing to online communities is not being familiar with the culture of that specific community first. Many communities (such as Reddit) dislike self-promotion, especially on newer accounts, so it's critical that you spend some time learning about the community and providing value before you share your extension. Below are a number of communities you can potentially share your extension on:

- Reddit: Spend some time finding the right subreddits for it to be effective; there are subreddits for just about anything (make sure to read individual subreddit rules first)
- Hacker News: Tends to be more focused on tech-related topics, but there is some room for other types of extensions (i.e. productivity, etc.)
- Product Hunt: Could be suitable for almost any type of extension; try to get a popular hunter to hunt your extension for you if possible

- [Indie Hackers](#): One of the best communities around for indie makers; very supportive of others and you'll often get great feedback on your product if shared
- [Facebook](#): Not necessarily Facebook itself, but the Groups that exist on it; there are many very niche groups with thousands of users on Facebook, it's just a matter of finding them

There are many more online communities that could be well-suited for sharing your extension as well. If your extension serves a more niche audience, there's likely separate communities around it as well.

## Directories

In addition to online communities mentioned above, there's also a wide variety of directories that you can post your extension to, such as [AlternativeTo](#), or [Alternative.me](#). If it's a general directory site, you can simply add your extension link with relevant details, however if it's a directory site that offers alternatives, it may require you to find a listed alternative to your extension first.

## Press

Press is often overlooked when talking about methods to gain exposure for your Chrome extension, but it can be highly effective. The easiest method to obtain press coverage on your extension is to find journalists who have already written about your niche or similar extensions to yours. Often times, if they've written about your niche in the past, they would gladly welcome new material to write about if you reach out and present your extension.

## Blogging

Blogging (sometimes referred to as content marketing) is another effective way to gain exposure for your extension. When blogging, you can either have your own blog, use a third-party blogging solution (like [Medium](#)), or do guest blogs. Regardless of the method you choose, you should focus on providing real value to your readers but still finding ways to integrate your extension into it in a way that makes sense and doesn't come across as too much self-promotion and not enough value provided. For example, if you have a productivity extension, you could write a blog post about the top 10 best productivity tools and list yours in there.

## Explainer Videos

Explainer videos can be a powerful tool as well. Not only does it provide a great way to show off your extension to potential users, it allows you to share it on other channels, such as YouTube or Vimeo, which may drive additional traffic organically from there.



## Advertising

Not necessarily something I would recommend if you don't have prior experience or if you want to keep your growth as low cost as possible, but paying to run some ads can sometimes have great returns, so I figured it was worth a mention. If you decide to go this route, do your research and build an effective ad strategy first.

## Listening to your users and stats

If you want to have a successful extension with continued growth over time, the development and general iteration process never really ends. It's important to constantly be looking at your stats and listening to your users in order to find areas for improvement and to fix bugs that come up. You can learn a lot from your users and stats if you know where and how to look for it.

### Users

Listening to your users is one of the most powerful tools you have when trying to build and grow a Chrome extension. The process of listening to your users never really comes to an end and requires you to constantly keep an eye out in order to iterate and improve where you see fit. With that said, not all feedback is created equal, so take the time to really make sure it's something that needs to be addressed before sinking time into it. The more you practice listening to user feedback and analyzing it, the better you'll get at it.

One of the best ways to get feedback from your users is through reviews on the Chrome extension store. Naturally, users will leave feedback on your extension in the form of a review over time and based on that you'll be able to not only tell if your extension is solving the problem like you intended, but you'll also often get bug reports and even some great feature requests.

Below are a couple reviews taken from one of my extensions, [Day Counter](#), that help show the extension is working as intended (names removed for privacy):



Modified Feb 10, 2018 ★★★★★

I have had this for one day and it has so far worked great. Thanks

Was this review helpful? ☐ Yes ☐ No [Reply](#) | [Mark as spam or abuse](#)

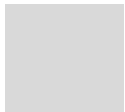


Modified May 18, 2018 ★★★★★

Perfect. Exactly what I needed. Thank you Ryan.

Was this review helpful? ☐ Yes ☐ No [Reply](#) | [Mark as spam or abuse](#)

However, the extension also had some reviews that reflected bug reports and as a result, I was able to tackle them and push an update fairly quickly:

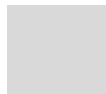


Modified Oct 4, 2017 ★★★★★

Great design but counters delete themselves when I close my tabs

Was this review helpful? ☐ Yes ☐ No [Reply](#) | [Mark as spam or abuse](#)

And probably best of all, I was able to receive a number of great feature ideas from users:



Modified Jan 21, 2018 ★★★★★

I would love the option of selecting WEEKS and DAYS eg 7weeks 4days (instead of just days).

Was this review helpful? ☐ Yes ☐ No [Reply](#) | [Mark as spam or abuse](#)



Modified Apr 16, 2018 ★★★★★

It would be amazing if the user could differentiate between the future and the past counters just by looking at them (like if the past counter's font turn red or something).

Was this review helpful? ☐ Yes ☐ No [Reply](#) | [Mark as spam or abuse](#)

As you can probably tell from the examples, there's a lot of value for you to act on in the reviews, so take advantage of it! Also, it's important to actively respond to users, especially those who are experiencing problems, as it will often result in their review being updated and improved which is better for your search rankings.

Outside of the reviews, it may also be worth reaching out to your users via email if you have access to it in order to obtain some feedback or even potentially using a third-party feedback collecting tool that you can add right into the extension itself where responses get sent directly to you.

## Stats

Stats tell you a lot about your extension and you can use what you learn from them to drive future development decisions. The Chrome Web Store provides stats for daily installs, daily uninstalls, daily impressions, and weekly users for every extension. The next chapter goes into depth about what each of these means and some insights you can draw from them.

# Understanding Extension Stats

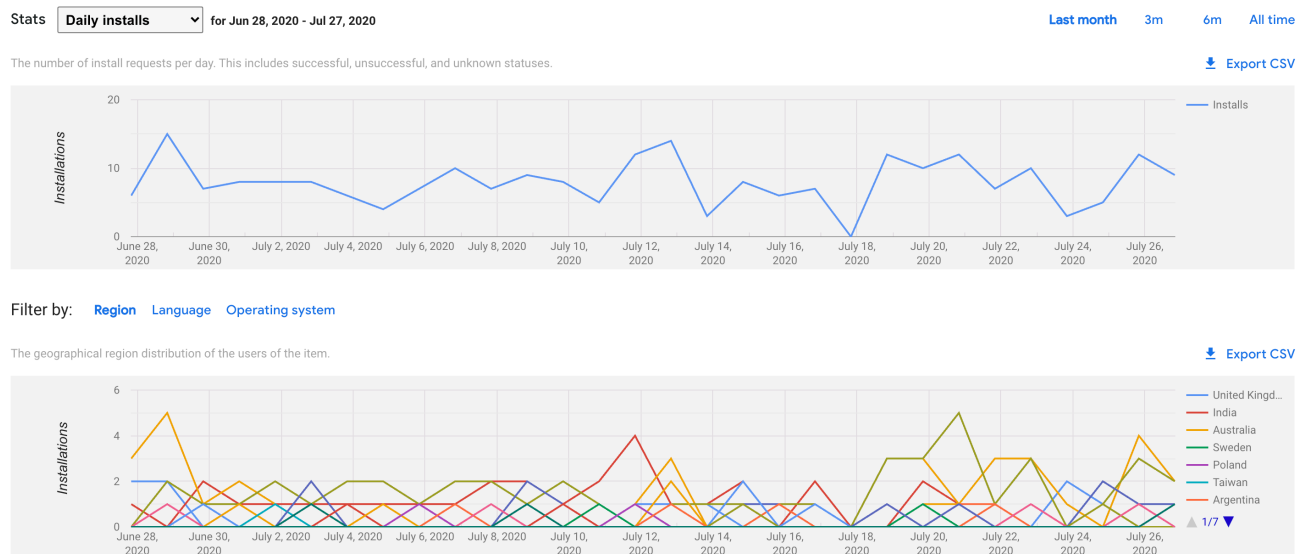
# Understanding Extension Stats

The Chrome Web Store Developer Dashboard provides 4 categories of stats for any published extension. Each category can provide some valuable insight into your extension's growth, so it's important to understand what each stat means and what it can tell you about your extension. Keep in mind, if your extension was recently published, it may take some time before the stats section can start providing meaningful insights to you.

## Daily Installs

The first stat provided is daily installs, which is the total number of requests to install your extension every day. It's important to note that this installation number is based on requests, which means successful, unsuccessful, and unknown statuses all count towards the number.

An example of the stats page for daily installs on *Day Counter* is as follows (filtered by *Last Month* at the time of writing):



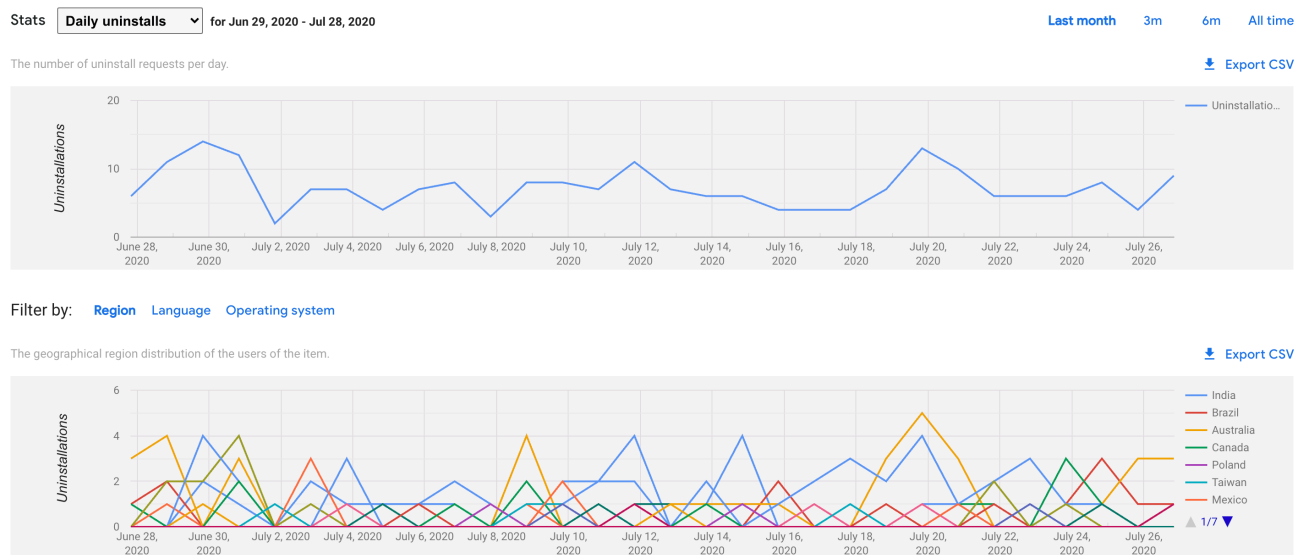
The top graph shows the installation request total for that time period and if you hover over a portion of the graph, you'll see the exact count for that data point. The graph below shows installation requests based on region, language, or operating system.

So how can this provide value? The daily installation count provides insight on the general growth of your extension. Using this particular stat, you can easily visualize how well your extension is growing over time. If the graph is trending down over time, you know growth is slowing. If the graph is trending up over time, you know growth is continuing. If growth is generally the same (like the screenshot above), you're still acquiring new users, but not at an increasing growth rate over time.

## Daily Uninstalls

The daily uninstalls stat tells you how many people request to uninstall your extension on per day over time. Similar to daily installs, it's based on requests and not necessarily successful requests alone.

An example of the stats page for daily uninstalls on *Day Counter* is as follows (filtered by *Last Month* at the time of writing):



Similar to daily installs, the top graph shows the uninstallation request total for that time period and if you hover over a portion of the graph, you'll see the exact count for that data point. The graph below shows uninstallation requests based on region, language, or operating system.

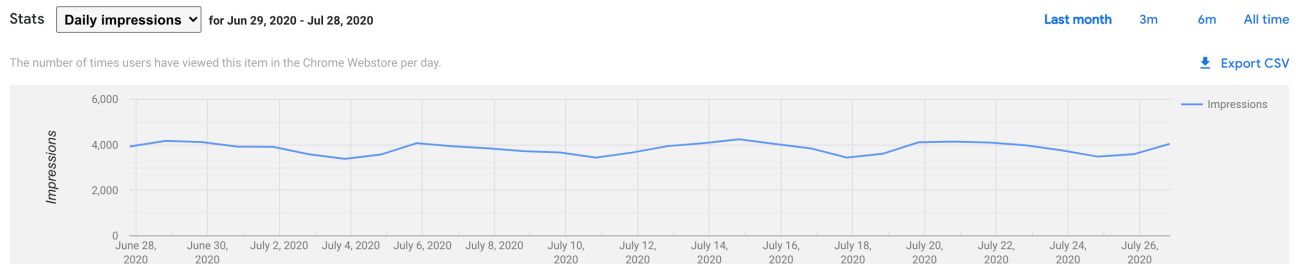
So how can this be valuable? The daily uninstallation count provides insight on how many users are removing your extension every day, similar to a churn rate of a SAAS business to a degree. You'll naturally get uninstalls on a daily basis as your extension grows, but it's important that your

number of install requests outnumbers your number of uninstall requests and that you don't have periods with huge numbers of uninstalls happening. Both the install and uninstall stats are more valuable when viewed in conjunction with each other. With that said, if you're experiencing large numbers of installs and equally large number of uninstalls, that's probably cause for some investigation. Without talking to your users, you can't necessarily say for sure, but common reasons for uninstalls are: users no longer using the extension, the extension is not what the users expected, or the extension isn't working correctly.

## Daily Impressions

The daily impressions stat tells you how many times users have viewed your Chrome extension's page in the Web Store per day. This can be thought of similarly to views on a website.

An example of the stats page for daily impressions on *Day Counter* is as follows (filtered by *Last Month* at the time of writing):



Unlike the install and uninstall stats page, the daily impressions stat page only has a single graph that charts the total extension page views per day over time.

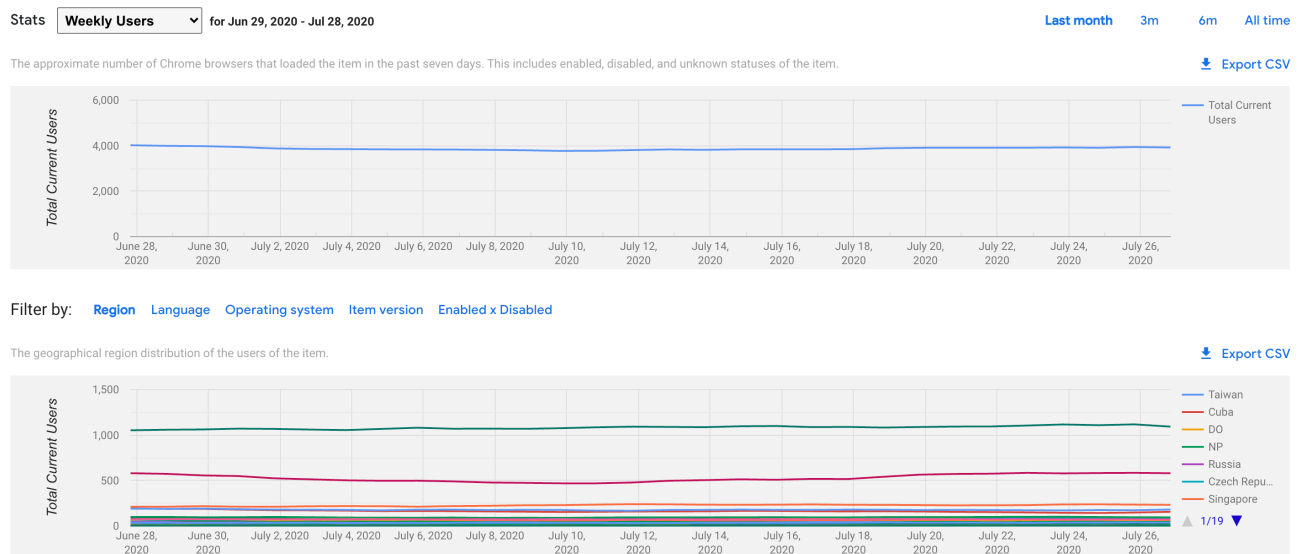
So how can this be valuable? Much like website views, knowing the views for your extension's page provides valuable insight as it can help you determine how effective the page is. For example, if you aren't getting many installs, but have a high volume of impressions, it likely means your extension page needs some attention to improve conversion (i.e. updating screenshots, tweaking the description, etc.).

One tip is to compare your impressions to your installs over time and after each new tweak to your extension page. By doing this, you can find which version of the extension page is converting users better. Another thing to note is you will likely not experience one-to-one for installs and impressions (in fact, your impressions will likely be far greater than installs per day), meaning not all users who view the page will install the extension.

## Weekly Users

The last stat provided in the dashboard is the weekly users, which displays how many Chrome browsers loaded your extension in the past seven days over the provided time period. This number includes enabled, disabled, and unknown statuses of the extension as well.

An example of the stats page for weekly users on *Day Counter* is as follows (filtered by *Last Month* at the time of writing):



Similar to the others, the top graph shows the number of loads in the last seven days for that time period and if you hover over a portion of the graph, you'll see the exact count for that data point. The graph below shows the same number based on region, language, operating system, version, and enabled vs disabled. Of particular interest in the bottom graph is the version filter and the enabled / disabled filter as it can help you figure out how many users are on each version of your extension and how many have it enabled / disabled, respectively.

So how can this be valuable? As you'd expect, this stat is really insightful in terms of showing you how many users are actually loading up and using your extension. Install count is valuable, but determining how many users are actually loading your extension is valuable as well because it helps you determine if your extension is being used and to what degree. If you have a good number of installs, but low weekly users, you know your extension isn't being utilized to the highest degree and can make tweaks to provide more value and increase that over time. Conversely, if you have high weekly users, you know your extension is being utilized.



Another valuable portion of this stat page is the item version filter because it shows you which versions of the extension each active user is on, which can help drive important decisions like backwards compatibility.