

Images haven't loaded yet. Please exit printing, wait for images to load, and try to print again.

Extension in 20 minutes



Jake Prins [Follow](#)

Feb 24, 2018 · 7 min read



Photo by Mikes Photos: <https://www.pexels.com/photo/acoustic-amplifier-artist-audio-114820/>

Ever wondered what it would be like to create a Chrome extension? Well, I'm here to tell you just how easy it is. Follow these steps and your idea will turn into reality and you'll be able to publish a real extension in the [Chrome Web Store](#) in no time.

What is a Chrome extension?

Chrome extensions allow you to add functionality to the Chrome web browser without diving deeply into native code. That's awesome

because you can create new extensions for Chrome with core technologies that web developers are very familiar with - HTML, CSS, and JavaScript. If you've ever built a web page, you'll be able to create an extension faster than you can have lunch. The only thing you need to learn is how to add some functionality to Chrome through some of the JavaScript APIs that Chrome exposes.

If you're not experienced yet in building web pages, I recommend you first dive into some free resources to learn how to code, like [freeCodeCamp](#).

What do you want to build?

Before you start, you should have a rough idea of what you want to build. It doesn't need to be some new groundbreaking idea, we can just do this for fun. In this article, I'll be telling you about my idea and how I implemented it into a Chrome extension.

The plan

I've used the [Unsplash](#) Chrome extension for a while which allows me to have nice background images of [Unsplash](#) in my default tab. I later replaced it with the [Muzli](#) Chrome extension that turns the default tab into a feed of design news and shots from around the web.

Let's use these two extensions as inspiration to build something new, but this time, for movie lovers. My idea is to show a random background image of a movie every time you open a new tab. On scroll it should turn into a nice feed of popular movies and TV shows. So let's get started.

Step 1: Setting things up

The first step is to create a manifest file named `manifest.json`. This is a metadata file in JSON format that contains properties like your extension's name, description, version number and so on. In this file we tell Chrome what the extension is going to do, and what permissions it requires.

For the movie extension we need to have permission to control **activeTab**, so our `manifest.json` file looks something like this:

```
{
  "manifest_version": 2,
  "name": "RaterFox",
  "description": "The most popular movies and TV shows in
your default tab. Includes ratings, summaries and the
ability to watch trailers.",
  "version": "1",
  "author": "Jake Prins",

  "browser_action": {
    "default_icon": "tab-icon.png",
    "default_title": "Have a good day"
  },

  "chrome_url_overrides" : {
    "newtab": "newtab.html"
  },

  "permissions": ["activeTab"]
}
```

As you can see, we say that `newtab.html` will be the HTML file that should be rendered every time a new tab gets opened. To do this we need to have permission to control the **activeTab**, so when a user tries to install the extension they will be warned with all the permissions the extension needs.

Add "RaterFox - Popular Movies & TV Shows"?

It can:

- Replace the page that you see when opening a new tab



Cancel

Add extension

Another interesting thing inside the `manifest.json` are the browser actions. In this example we use it to set the title, but there are more options. For instance, to show a popup whenever you click on the app icon inside the address bar, all you have to do is something like this:

```
"browser_action": {
  "default_popup": "popup.html",
},
```

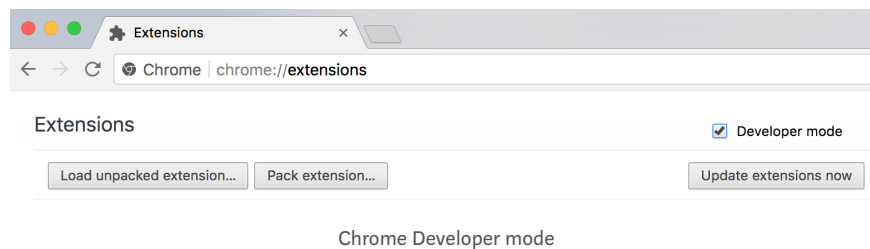
Now, `popup.html` will be rendered inside the popup window that's created in response to a user's click on the browser action. It's a standard HTML file so it's giving you free reign over what the popup displays. Just put some of your magic inside a file named `popup.html`.

Step 2: Test if it works

The next step is to create the `newtab.html` file and put in a 'Hello world':

```
<!doctype html>
<html>
  <head>
    <title>Test</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

To test if it works, visit `chrome://extensions` in your browser and ensure that the **Developer mode** checkbox in the top right-hand corner is checked.



Click **Load unpacked extension** and select the directory in which your extension files live. If the extension is valid, it will be active straight away so you can open a new tab to see your 'Hello world'.

Step 3: Making things nice

Now that we got our first feature working, it's time to make it nice. We can simply style our new tab by creating a `main.css` file in our extension directory and load it in our `newtab.html` file. The same goes when including a JavaScript file for any active functionality that you

would like to include. Assuming that you have created a web page before, you can now use your magic to show your users whatever you want.

Finishing up the plan

All I further needed to finish the movie extension was HTML, CSS and JavaScript, so I don't think it's relevant to dive deep into the code, but I'd like to go through it quickly.

Here is what I did:

For my idea I needed some nice background images, so in the JavaScript file I used the [TMDb API](#) to fetch some popular movies, took their backdrop images and put them in an array. Whenever the page loads it now randomly picks one image from that array and sets it as the background of the page. To make this page a little more interesting I also added the current date in the top right corner. And for more information, it allows users to click the background which leads to visiting the movie's [IMDb](#) page.

I replaced the screen with a nice feed of popular movies when the user tries to scroll down. I used the same API to build cards of movies with an image, title, rating and vote count. Then, on clicking one of those cards, it shows the overview with a button to watch a trailer.

The result

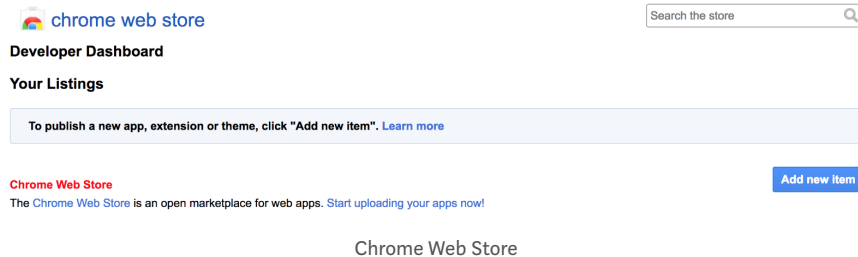
Now with that little `manifest.json` file and just some HTML, CSS and JavaScript, every new tab that you open looks a lot more interesting:



View the end result here.

Step 4: Publish your extension

When your first Chrome extension looks nice and works like it should, it's time to publish it to the Chrome Store. Simply follow [this link](#) to go to your Chrome Web Store dashboard (you'll be asked to sign in to your Google account if you're not). Then click the **Add new item** button, accept the terms and you will go to the page where you can upload your extension. Now compress the folder that contains your project and upload that ZIP file.



After successfully uploading your file, you will see a form in which you should add some information about your extension. You can add an icon, a detailed description, upload some screenshots, and so on.

Make sure you provide some nice images to show off your project. The store can use these images to promote your groundbreaking project. The more images you provide, the more prominently your extension will be featured. You can preview how your extension looks inside the web store by clicking the **Preview changes** button. When you're happy with the result, hit **Publish changes** and that's it, your done!

Now go to the [Chrome Web Store](#) and search for your extension by its title (It might take some time before it's up there). If you're interested, you can find mine [here](#).

The only thing left to do is get some users. So you might want to share a post about your life changing Chrome extension on social media. Tell your friends to check it out. Add it to [ProductHunt](#). And don't forget to share your project here in the comments. I'm curious to see what you came up with!

Conclusion

As a web developer, it's very easy to create a Chrome extension in a short amount of time. All you need is some HTML, CSS, JavaScript and a basic knowledge of how to add functionality through some of the JavaScript APIs that Chrome exposes. Your initial setup can be published inside the Chrome Web Store within just 20 minutes. Building an extension that's new, worthwhile or looks nice will take some more time. But it's all up to you!

Use your creativity to come up with something interesting and if you ever get stuck, the excellent Chrome extension [documentation](#) can probably help you out.

So what are you waiting for? It's time to start working on your own Chrome extension and turning your idea into reality.

Don't forget to share your project in the comments and hit the clap button if this article was any useful to you. If you got some time and want to be a hero, give [my extension](#) a positive rating. That would be highly appreciated!

Got questions or feedback? Let me know in the comments!

. . .

Thanks for reading! Hope the information was helpfull. Follow me on Medium for more tech related articles or on Twitter and Instagram [@jakeprins_nl](#).

