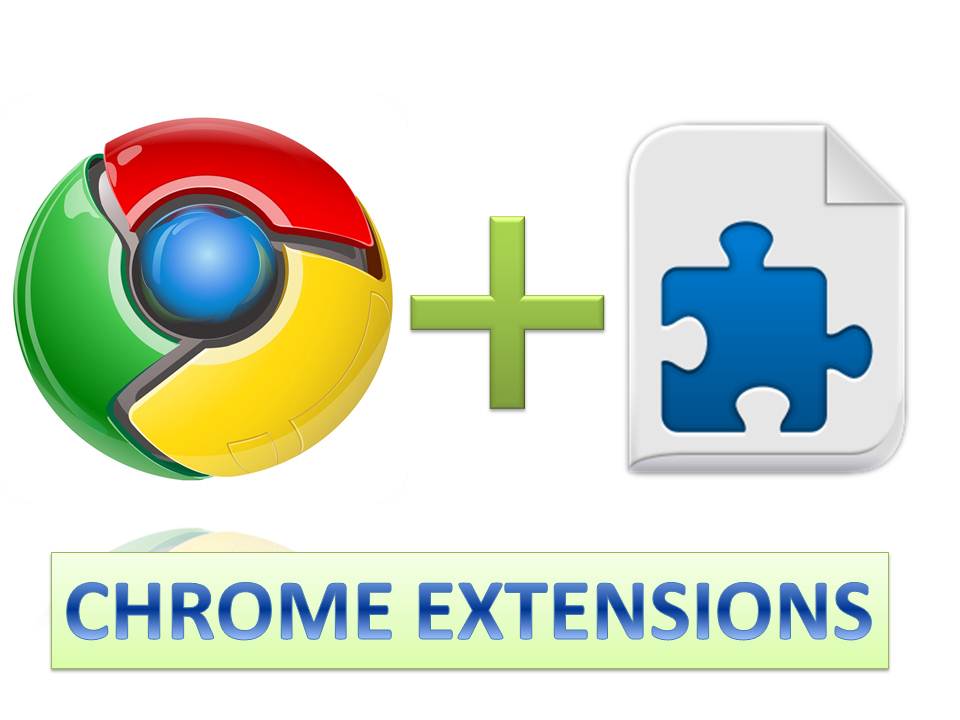
**WORKING WITH GOOGLE CHROME EXTENSIONS**



**INTRODUCTION**

Chrome is an awesome browser from Google that is very fast and lightweight, yet also very powerful. Chrome also has a very good extensibility model that allows developers with just HTML, CSS and JavaScript skills to create powerful extensions. In this document you will get to know about some basics of Chrome extensions and how extensions work in chrome and finally you will see how easily we can create chrome extensions.

**WHAT IS CHROME EXTENSION?**

Extensions are small software programs that can modify and enhance the functionality of the Chrome browser. You write them using web technologies such as HTML, JavaScript, and CSS. It can be thought of a JavaScript application that browser runs and invoked at certain points that developer of the extension can specify. Extensions bundle all their files into a single file that the user downloads and installs. This bundling means that, unlike ordinary web apps, extensions don't need to depend on content from the web.

**EXTENSION VERSUS PLUGIN**

Extensions are often confused with plugins which are quite different. Many developers mistakenly confuse them as the same, but they are not. There is quite a bit of debate across the web about the actual differences, but the easy way to conceptualize it is to think of Plugins as running outside of the context of the browser and interacting with the browser. Extensions on the other hand, run within the browser and conform to a specific set of APIs and capabilities exposed by Chrome.

**WHAT EXTENSION CAN DO**

* **Interacts with webpages or servers.**

This means that you can have an extension interact with APIs or other services to perform its functionality. You aren't restricted to just what you can package in your extension file.

* **Control browser features.**

Through Chrome APIs you can access things like font settings, tabs, downloads, and even browser history.

* **Adding user interface to chrome.**

This means you can add custom UI to chrome through:

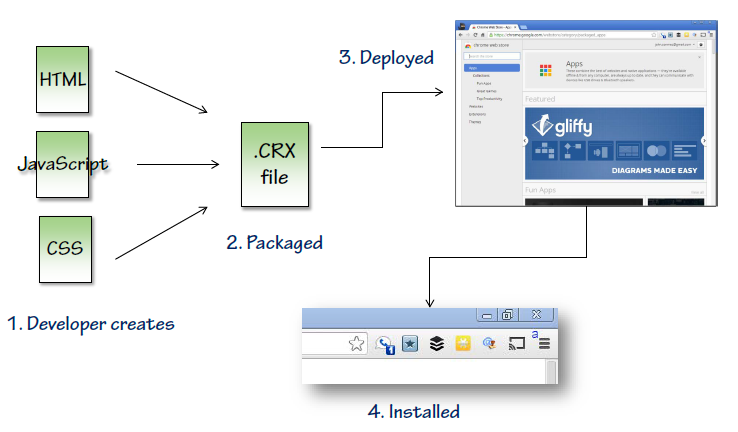
* + **Browser Action:** Through chrome browser actions a button can be added to the chrome toolbar that can be pressed to display a custom user interface.
  + **Page Action:** Page actions can put icons in the address bar that can be clicked to access a user interface for actions specific to that page.
* **Create options page.**

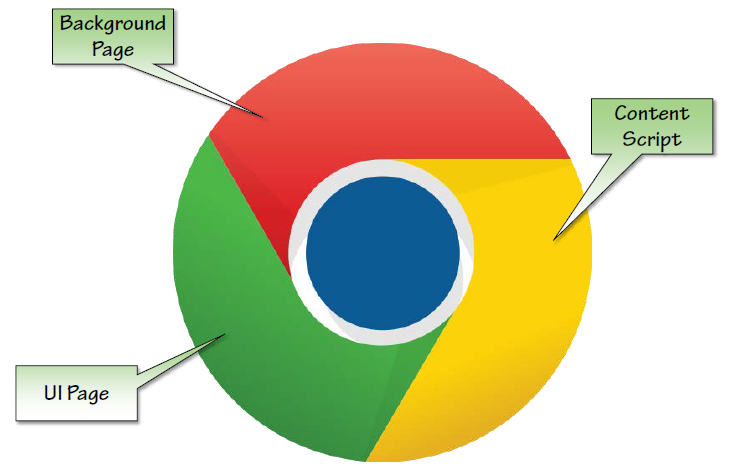
Extensions can have its own options page where it can be configured through own custom UI.

* **Modify loaded pages.**

Extensions can modify loaded pages by injecting JavaScript into the page to add capabilities to the page or even change the UI of it.

**HOW CHROME EXTENSIONS WORK**





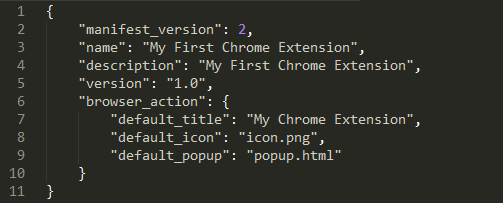
It uses common web technologies like Html, CSS, and JavaScript to extend the capability of chrome browser which gets bundled together to create one .CRX file which gets deployed in chrome browser.

**BUILDING MY FIRST CHROME EXTENSION**

**STEP 1:** Create a new folder which will contain a “***manifest.json***” file which will describe the properties of extension and also information for chrome web store, icon file which would be an image that will show in chrome toolbar to highlight and trigger the extension and a html page that will display when extension icon is clicked. **Note:** Manifest version must be **“2”** as version **“1”** is been deprecated since Chrome 18.0 by Google due to some content security policy.

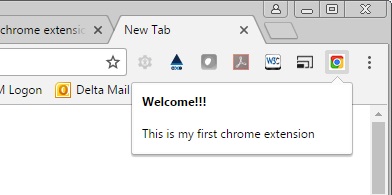


**STEP 2:** Your *manifest.json* file will look like this :



**STEP 3:** Now basic folder setup is done for your extension. Now to add extension to chrome browser you need to proceed in this manner:

Open chrome browser 🡪 open this URL ***chrome://extensions/*** 🡪 Check the ***developer mode*** checkbox 🡪 Click on ***Load unpacked extension*** button 🡪 Select extension folder 🡪 Click ***OK***



After this you are all set. To activate your extension, click on the extension icon from chrome toolbar. This will open your popup.html as shown above.

**STEP 4:** So now your basic chrome extension is ready. So let us see how interact with rendered pages using chrome extension. For this we need to make some changes in manifest.json file and to allow some permissions. Your modified manifest.json will look like this:



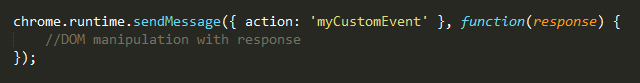
So, we have made three changes in manifest.json file, i.e., “content\_scripts”, “background” and “permissions”. Now there will be one JS file for the extension which in this case is “**background.js**”, that will interact with my content\_script which is “**content.js**” and this content.js will contain all the code to manipulate the rendered pages. Background.js will handle backend related tasks like http calls, data processing and to send the processed data to content script that can be used to do some interactions with the UI.

**Permissions:** In permissions we need to specify the URLs that our extension will use for some backend call to fetch data.

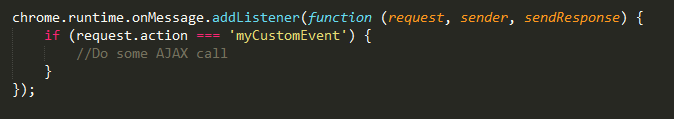
**INTERACTING WITH SOME CHROME APIs**

In “content\_script”, there is a key called “matches” whose value is a URL, which specifies that our content.js will be loaded for the pages specified by the URL.

Now let’s say we need to do an ajax call after content script fires, and as discussed all our backend related scripts (Other than DOM manipulation) will be in “background.js” file. So on page load we will emit an event using the Chrome API as shown below:



After that the event called **“myCustomEvent”** will be captured in “background.js” as shown below:



**So basically we emit an event via chrome API when the page loads and capture it in background page and did some data processing**.

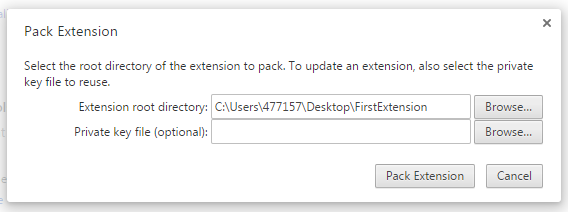
**CREATING .CRX FILE / PACKAGING**

Packaging is basically required if you want to publish your extension using Chrome Developer Dashboard by creating **“.crx”** file. You can also upload your extension file at Chrome Web Store so that people can download and can use your extension. To package an extension:

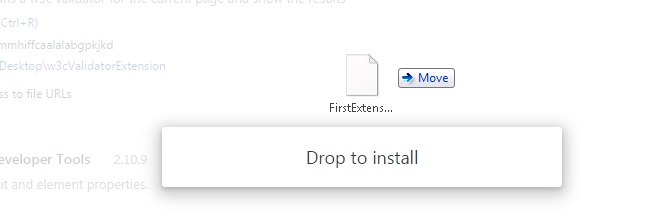
1. Bring up the extensions management page by going to this URL:

***chrome://extensions***

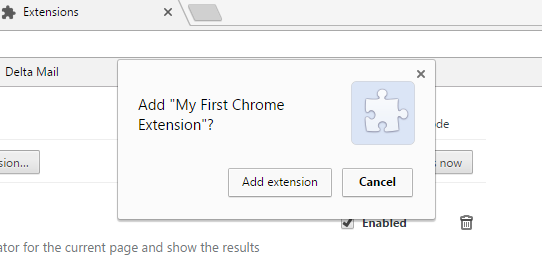
1. Ensure that the "Developer mode" checkbox in the top right-hand corner is checked.
2. Click the Pack extension button. A dialog appears.
3. In the Extension root directory field, specify the path to the extension's folder as shown:



1. Click Pack Extension. The packager creates two files: **.crx** file, which is the actual extension that can be installed, and a **.pem** file, which contains the private key.
2. Now simply drag and drop your **.crx** file in your google chrome extensions.



1. A dialog appears. Click on Add Extension and its done.



**REFERENCES**

[*https://developer.chrome.com/extensions*](https://developer.chrome.com/extensions)

[*https://developer.chrome.com/extensions/api\_index*](https://developer.chrome.com/extensions/api_index)

[*https://developer.chrome.com/extensions/browserAction*](https://developer.chrome.com/extensions/browserAction)

[*https://developer.chrome.com/extensions/packaging*](https://developer.chrome.com/extensions/packaging)

**Thanks & Regards,**

**Yatin Aggarwal (477157)**

CDBINT