Coding Part

11_Bugs

October 13, 2019

Writing a Chrome extension:

Chrome extensions are build using standard web technologies. The primary differences between an extension and a web app are that extensions have access to additional browser specific APIs and run differently that normal pages.

To start off, create a new folder with the name of your extension and add the following file "manifest.json" to it. The manifest is an "index" of the extension and most of the fields are self explanatory.

```
{
"name":  "Ad-blocker",
"version":  "1.0",
"description":  "A simple adblocker",
"permissions":  ["webRequest", "webRequestBlocking", "http://*/",
"https://*/"],
"background":  {
"scripts":  ["blocked_domains.js", "background.js"]
},
"manifest_version":  2
}
```

The file manifest.json lists the components of the extension and describes what permissions it needs. The permissions field lists what webpages the extension has permission to access. We want our extension to have access to all pages, hence the wildcard patterns. Note: permission to access an http page is distinct from permission for the https equivalent. The permission field also lists which "special" APIs and feature we wish to make use of.

Our extension will run in the background and automatically intercept and filter requests. To set this up we give a list of scripts to load and run in the background. The order of scrips is important so make sure to list a script after its dependencies.

webRequests API:

The plan for the blocker is to intercept requests and block ones to domains hosting ads. Perusing the chrome docs reveals the chrome.webRequest.onBeforeRequest event. The onBefore request "Fires when a request is about to occur. This event is sent before any TCP connection is made and can be used to cancel or redirect requests. background.js sets up an event handler to cancel all requests to the specified urls.

Creating a blacklist:

Now edit the blocked_domains.js so that it is a valid JavaScript array that looks like this:

```
var blocked_domains = [
"*://*.lb.usemaxserver.de/*",
"*://*.tracking.klickthru.com/*",
...
...
"*://*.zmedia.com/*",
"*://*.zv1.november-lax.com/*"];
```

blocked_domains.js consists of a single array of blacklisted domains Also our background.js will look like this.

```
    chrome.webRequest.onBeforeRequest.addListener(
function(details) {
console.log("blocking:", details.url);
return {cancel:  true };
},
{urls:  blocked_domains },
   blocking"

);
```

Adding UI:

There is a possibility that this extension could break the main functionality of some sites. So let's add an enable/disable button. The first step is to update the manifest to specify the location of the html page that will show when the extension's icon is clicked on.

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

Add the following entry to the manifest to specify the control panel page Now create fill out popup.html with a toggle button. Chrome does not like inline JavaScript in extension pages, so the JavaScript to handle button clicks will be in a separate file.

```
<html>
<head>
<script src="popup.js"></script>
</head>
<body>
<h3>Ad-blocker</h3>
<input type="button" id="toggle_button" value="Disable" />
<hr>
Written by Bugs
</body>
</html>
```

Not much here. Just a header, button, and script.

Finally...

Final version of background.js

```
var enabled = true;
chrome.webRequest.onBeforeRequest.addListener(
function(details) {
return cancel:  enabled ;
},
{urls:  blocked_domains },
   blocking"

);  popup.js
window.onload = function () {
function updateLabel() {
var enabled = chrome.extension.getBackgroundPage().enabled;
document.getElementById('toggle_button').value = enabled ?  "Disable"
:  "Enable";
}
document.getElementById('toggle_button').onclick = function () {
var background = chrome.extension.getBackgroundPage();
background.enabled = !background.enabled;
updateLabel();
};
updateLabel();
}
```

Note: So there we have it, a short simple adblocker. The only major problem is keeping the blacklist up to date, but scripts could make that easy.