



# Coding Dojal - Browser Extensions

Jean-François Grand



@gritchou



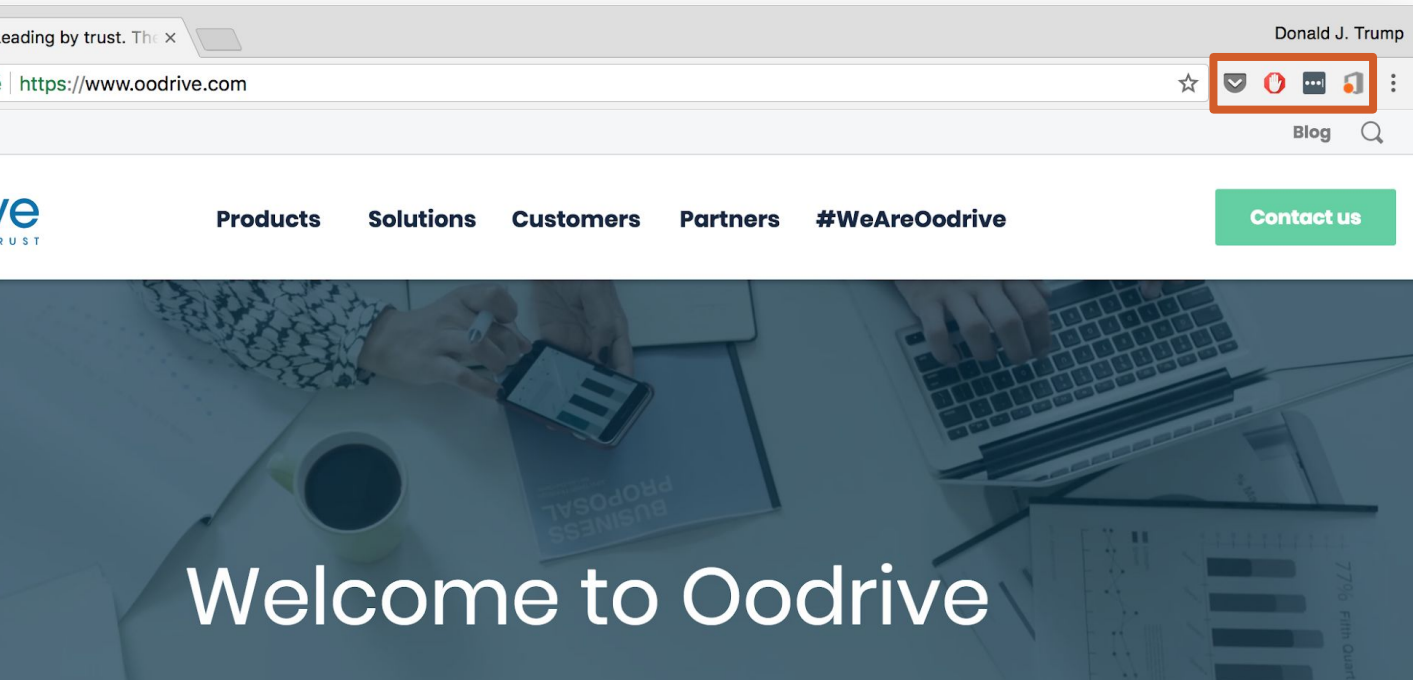
# Summary

## Browser extensions?

1. Hello World
2. Background
3. API
4. Options page
5. Popup
6. Content script



# What are Browser Extensions?



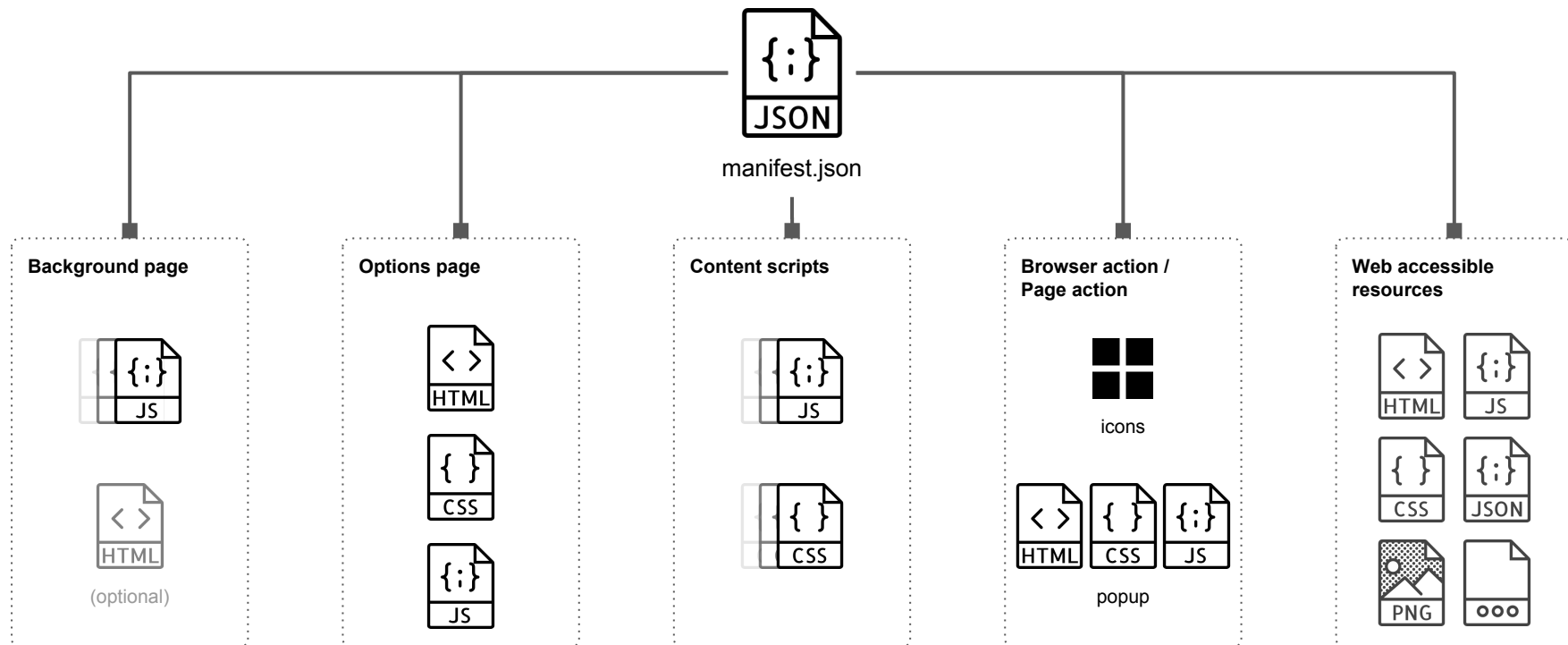
# Dissecting a Browser Extension (alive!)

## Structure of manifest.json

```
1 {  
2   "manifest_version": 2,  
3   "name": "Translator",  
4   "version": "0.0.1",  
5  
6   "permissions": ["activeTab", "background", "contextMenus"],  
7  
8   "background": {  
9     "scripts": ["jquery.js", "background.js"],  
10    "persistent": false  
11  },  
12  
13  "browser_action": {  
14    "default_icon": {  
15      "19": "icons/logo-19.png",  
16      "38": "icons/logo-38.png"
```

activeTab, background, contextMenus, jquery.js, background.js, permissions, browser\_action, default\_icon, icons/logo-19.png, icons/logo-38.png, manifest\_version, name, version, enterprise.deviceAttributes, topSites, clipboardRead, fontSettings, sessions, webNavigation, system.cpu, idle, platformKeys, pageCapture, tts, notifications, system.memory, printerProvider, proxy, tabCapture, management, ttsEngine, storage, certificateProvider, idltest, webRequest, bookmarks, experimental, cookies, alarms, clipboardWrite, signedInDevices, debugger, tabs, displaySource, system.display, vpnProvider, identity, background, privacy, enterprise.platformKeys, history, unlimitedStorage, declarativeContent, documentScan, gcm, fileBrowserHandler, system.storage, webRequestBlocking, desktopCapture, networking.config, contentSettings, downloads, fileSystemProvider

# Dissecting a Browser Extension (alive!)



# API

alarms extension windows browserAction contentScripts history clipboard  
management tabs permissions  
downloads browserSettings contextMenus notifications  
webRequest commands runtime proxy  
storage bookmarks devtools.panels

# API

management  
contextMenus  
webRequest  
storage  
downloads  
tabs  
history  
browserSettings  
proxy  
clipboard

# Chrome



Namespace `chrome.*` with callback functions

```
chrome.tabs.query({ active: true }, (tabs) => {  
  ...  
});  
  
chrome.tabs.query({ active: true }, function(tabs) {  
  ...  
});
```

- [Browser Extensions MDN](#)
- [Chrome API](#)



# Repository

```
chrome.tabs.query({ active: true }, (tabs) => {  
  ...  
});  
  
chrome.tabs.query({ active: true }, function(tabs) {  
  ...  
});
```

- [Browser Extensions MDN](#)
- [Chrome API](#)

# Exercise 1 - Hello World

Display "hello world" in a popup

## Requirements

---

- Have a cool icon (owl.png 128px)
- Display some cool text ("Where are my cookies Louis?")
- Set homepage url
- Add a keyboard shortcut to open popup

## Useful Links

---

- [Browser Extensions on MDN](#)
- [manifest.json](#)
- Manifest Entry [browser\\_action](#)
- Manifest Entry [homepage\\_url](#)
- Manifest Entry [commands](#)

# Exercise 2 - Background Script & Context Menu

Search selected text on giphy

## Requirements

---

- Add a Context Menu entry
- Context Menu entry must be updated with selected text
- Open new Tab with giphy search

## Useful Links

---

- [Giphy!](#)
- Manifest Entry [background](#)
- Manifest [permissions](#)
- Chrome [Tabs API](#)
- Chrome [Context Menu API](#)

# Exercise 3 - From Giphy API to clipboard

Transform selected text to clipboard gif

## Requirements

---

- Get an API key from giphy
- Get background page from chrome extension API
- Add a textarea or input to the page to interact with clipboard
- Take only first result returned by API
- Spam rocket chat channels with gifs
- Enjoy

## Useful Links

---

- [Giphy API](#)
- Manifest [permissions](#)
- [Send method from XMLHttpRequest](#)
- [chrome.extension API](#)
- [Create an Element in Document](#)
- Interact with the [clipboard](#)

# Exercise 4 - Options Page

Create an options page to customize giphy api call

## Requirements

---

- Customize number of results wanted
- Options must open in a separate tab
- Options must be stored using storage api
- Options must be persisted in options page
- Allow switch between Search and Random giphy api

## Useful Links

---

- Manifest Entry [options\\_ui](#)
- [Chrome storage API](#)
- [input HTML Element](#)
- [radio HTML Element](#)

# Exercise 5 - Browser Action Popup

Create a popup that displays Gifs and has a button to let user copy to clipboard

## Requirements

---

- Word to search will be stored in an input in the options page
- Runtime API should be used to discuss with the popup
- Number of results to display will be stored in the options page aswell
- One button per gif will let user copy url to clipboard

## Useful Links

---

- [Using fetch](#)
- Manifest Entry [browser\\_action](#)
- [button HTML Element](#)
- [Runtime sendMessage](#)
- [Runtime onMessage](#)

# Exercise 6 - Content Script

Replace all image in a website by gifs

## Requirements

---

- Free For All

## Useful Links

---

- None

# Further reading

- [Browser Extensions on MDN](#)
- [Mozilla \*\*polyfill\*\* to help cross-browser extension development](#)
- [CLI to help scaffolding a cross-browser extension project](#)
- [Icebrg blog article on malicious chrome extensions](#)
- [Google security blog](#)
- [Source code for the Translator browser extension](#)



# Thank you!

## Notice and acknowledgments

All product names, logos, and brands are property of their respective owners. All company, product and service names used in this presentation are for identification purposes only.

Icons made by [Eucalypt](#), [Freepik](#), [Kiranshastry](#), [Smashicons](#), [Icomoon](#), [dave-gandy](#) from [flaticon.com](#).

Icons made by [mikicon](#) from [thenounproject.com](#).