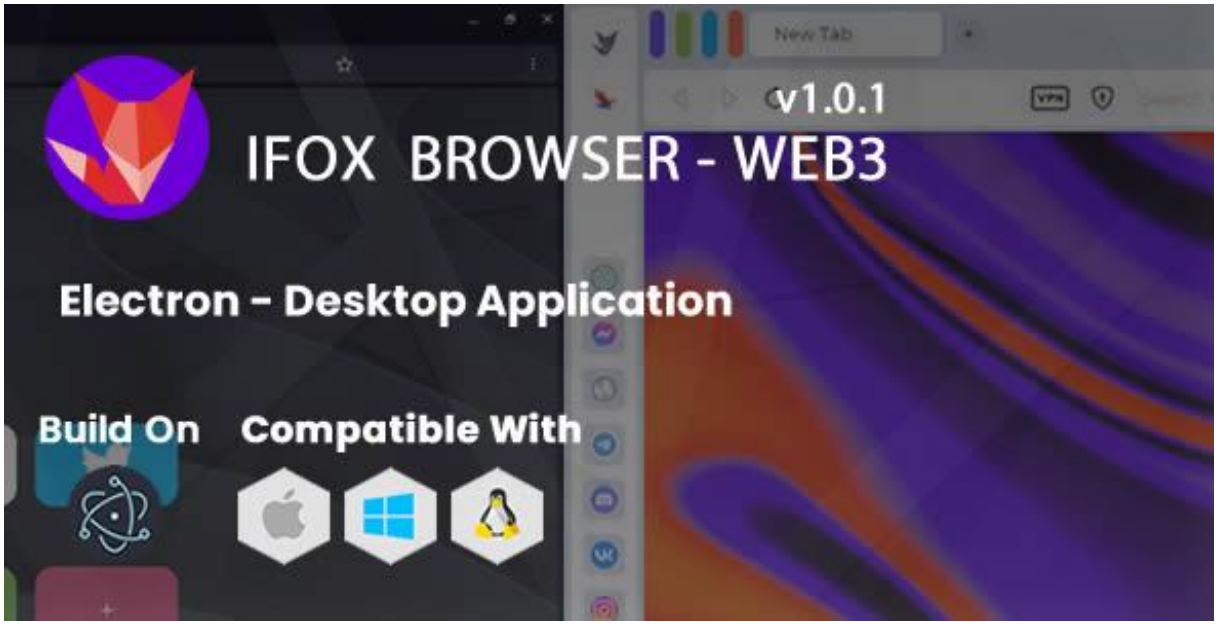
IFOX Browser - Web3



It is a fast and secure web browser for your Windows, Linux, Mac OS.

Created: 3/11/2020
Latest update: 28/07/2023
BY: Mert Çetin
Support Platform : https://github.com/cetinmert/IFOX/issues
Requirements : -Node JS: v18+ | -Npm: v7+ | Git

✓ J Join our Official Github issues for more discussion and hints, bugs, fixes, and anything related to Mobile applications. ✓ Join our community on Telegram.

With IFOX, macOS, Windows and Linux you can create your own browser application in minutes.

Desktop application created with IFOX Electron;

Build cross-platform desktop apps with JavaScript, HTML, and CSS

- Build using Electron framework
- Easy to customize and edit
- Cutom menu options
- Support for MacOS, Windows, Linux
- Easy to build an application for your preferred OS

- More features will be adding soon

Installation Guide

The installation is pretty easy, please follow the steps below:

- IFOX Web Browser Script, you can get it from Here
- Download Node.js - Install Node.js Here
- Download Git - Install Git Here

- Unzip the IFOX archive, extract it to new folder, and then open the folder.
- In the main folder you will find the solution.
- Extract IFOX-browser-1-0-1.zip to your workspace

install node.js

First, we install node.js on our computer. For installation, you can download the version suitable for your computer from the link below.

https://nodejs.org/en/download

With the installation of Node.js, the npm command will be activated in your computer's command client. To check, you can check it by typing `npm -v` in the command prompt.

Installing *Git* – the easy way

https://gist.github.com/derhuerst/1b15ff4652a867391f03

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
– The *Git* HYPERLINK "http://git-scm.com/" website

Choose one of the following options.

- Instructions for *HYPERLINK "https://gist.github.com/derhuerst/1b15ff4652a867391f03"Windows*

- Instructions for *HYPERLINK "https://gist.github.com/derhuerst/1b15ff4652a867391f03"*Mac
- Instructions for *HYPERLINK "https://gist.github.com/derhuerst/1b15ff4652a867391f03"*Linux

package.json Editing

Open to edit package.json;
If automatic updating will be enabled, it is necessary to connect a github repo. (The github repo will only contain all packaged applications.)

"repository": {
"type": "git",
"url": "https://github.com/github_usename/reponame.git"
},

"publish": [
{
`"provider": "github",`

`"owner": "github_usename",` -> Github User Name

`"repo": "repo_name",` -> Github Repo Name

`"vPrefixedTagName": true,`

`"releaseType": "draft",` -> Github send as a draft

`"publishAutoUpdate": true,`

`"token": "github_token"` -> Secret token to connect to Github
}
],

To publish the packaged applications

--publish always -> To Publish
--publish never -> To Not Publish

```
"scripts": {
"start": "electron .",
"build-linux": "electron-builder --linux --publish never",
"build-win": "electron-builder --win --publish always",
"build-mac": "electron-builder --mac --publish never"
},
```

Both of the build commands use electron-builder and output executables for their respective operating system in the dist/ folder.

Download and initialize IFOX source code with these commands:

- Open command prompt (CMD), PowerShell or terminal
- cd IFOX -> Get the directory path from which we extracted the project and run it
- npm i -> npm include in the Project

Run IFOX from source:

npm run start

Build instructions for Windows:

npm run build-win

Build instructions for Linux:

npm run build-linux

Build instructions for Mac:

npm run build-mac

Files Structure

- node_modules - All dependencies included in this folder (Electron).
- css folder - Images used on other pages, js - css files included in this folder.
- build folder – Include all app icons.
- images folder - All used Images.
- js folder - All used javascript.
- static/pages folder - all pages used
- translations folder - language files, translations
- dist folderBuilt package files, applications are here
- main.js - Application configuration file. (Application Name, application window sizes and more.)
- package.json - NodeJS application and package details.

Customizing and Configurations

Change application Name

First, you need to change the name attribute of the package.json file in the application root directory .

```
"name": "IFOX",
```

Next, modify config.js file appName value.

```
"productName": "IFOX",
```

Change Application Title Name

Change application description

First you need to change the application root directory package.json description attribute.

```
"description": "IFOX Web Browser by Mert Çetin for Me Force",
```

Google Settings, Location Connection -

> https://console.cloud.google.com/

First, you need to change the process.env attribute of the main.js file in the application root directory.

```
process.env.GOOGLE_API_KEY = 'XXxxXxXXxXX00X0xxx00XXxXx0Xx-xxXXXXxXXx';
process.env.GOOGLE_DEFAULT_CLIENT_ID = '000000000000-
x00000xxxx0x000xxx0xxxxxxxxxxx0.apps.googleusercontent.com'
process.env.GOOGLE_DEFAULT_CLIENT_SECRET = 'XxxxxXxxX00xx0XXxXxx0xxxX'
```

Change the title, width and height

Open your application main.js file, after that change the title, height, width, minHeight, and minWidth values.

mainWindow = new BrowserWindow

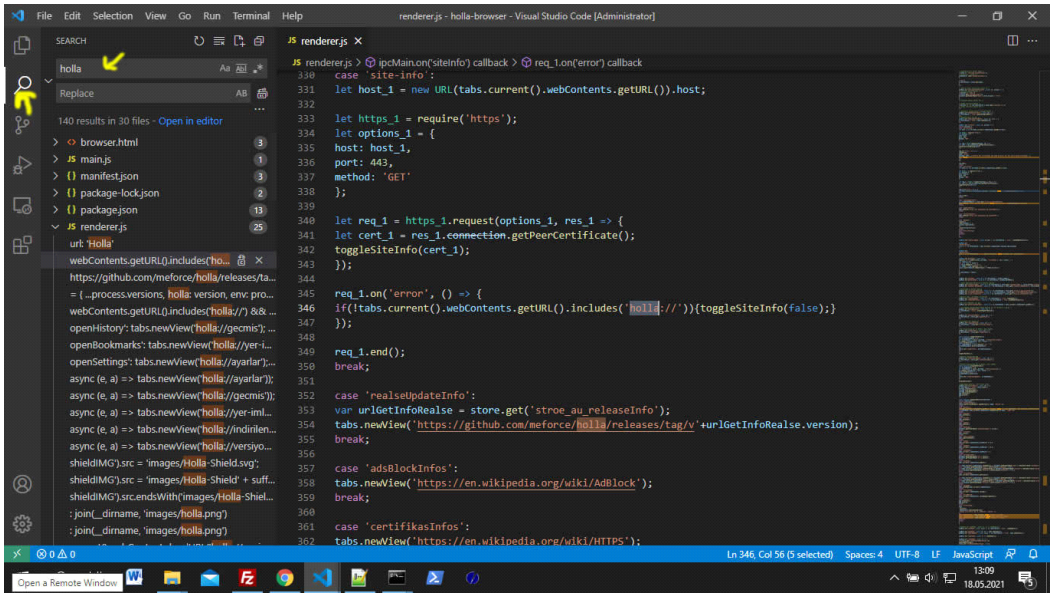
```
title: 'IFOX',
minWidth: 500,
minHeight: 450,
backgroundColor: '#FFFFFF',
width: 1280,
height: 720,
```

Recommended: do not change width and height settings

Change the Home Router IFOX: path

As in Chrome, there is a main router in chrome:// IFOX.
Changing this router address is easy.

- First open project with visual studio code
- In the top right menu, File -> Open Folder
- Search for IFOX and IFOX:// from the search section



Sections to Change

Find these and replace only the IFOX part

```
ifox://
```

```
startsWith('ifox')
```

```
window.location.protocol == 'ifox:'
```

Find all replace is easy. The best method is to check and replace them one by one.

Important

- Don't forget to take a backup before starting the process.
- Do not modify the main project files. Example (browser.html, main.js, package.json, node_modules etc).

Code Signing

Code signing is a security technology that you use to certify that an app was created by you.

On macOS the system can detect any change to the app, whether the change is introduced accidentally or by malicious code.

On Windows, the system assigns a trust level to your code signing certificate which if you don't have, or if your trust level is low, will cause security dialogs to appear when users start using your application. Trust level builds over time so it's better to start code signing as early as possible.

While it is possible to distribute unsigned apps, it is not recommended. Both Windows and macOS will, by default, prevent either the download or the execution of unsigned applications. Starting with macOS Catalina (version 10.15), users have to go through multiple manual steps to open unsigned applications.

Electron Documentation https://www.electronjs.org/docs/tutorial/code-signing

Sign your installer package to avoid smartscreen filtering.

https://github.com/electron/windows-installer#sign-your-installer-or-else-bad-things-will-happen

Sign your installer or else bad things will happen

For development / internal use, creating installers without a signature is okay, but for a production app you need to sign your application. Internet Explorer's SmartScreen filter will block your app from being downloaded, and many anti-virus vendors will consider your app as malware unless you obtain a valid cert.

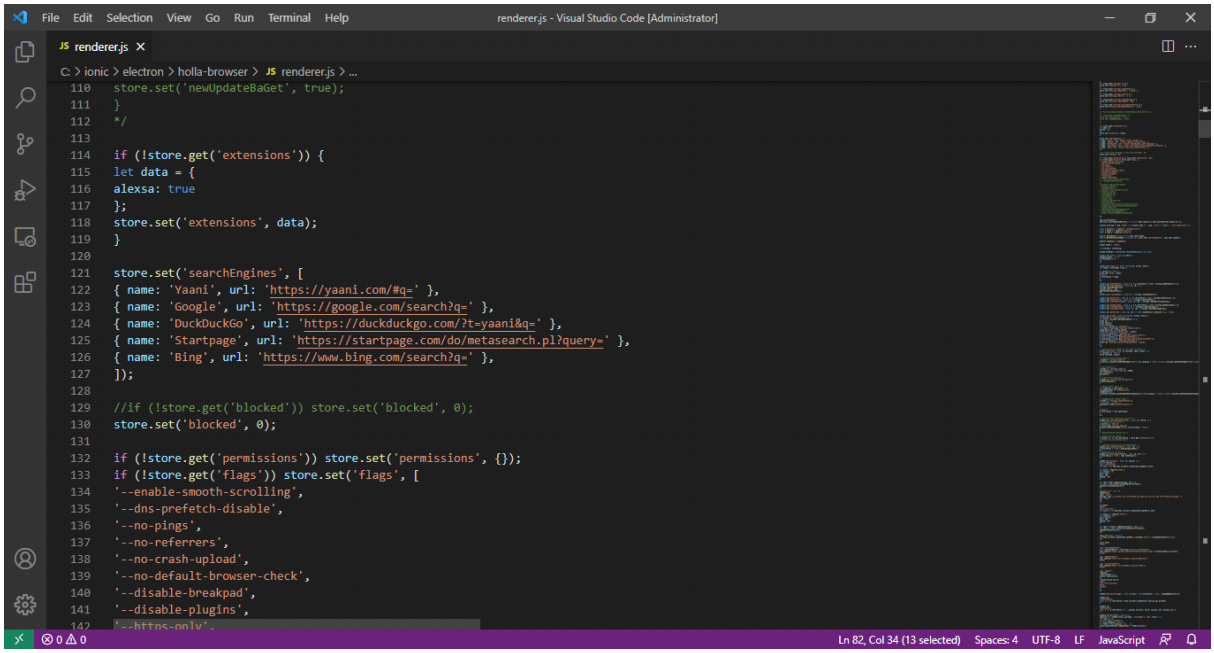
Any certificate valid for "Authenticode Code Signing" will work here, but if you get the right kind of code certificate, you can also opt-in to Windows Error Reporting.. This MSDN page has the latest links on where to get a WER-compatible certificate. The "Standard Code Signing" certificate is sufficient for this purpose.

Changing Search Engine And Adding New

Changing a search engine or adding a new one is pretty easy.

- First open project with visual studio code
- Open renderer.js file

121 lines

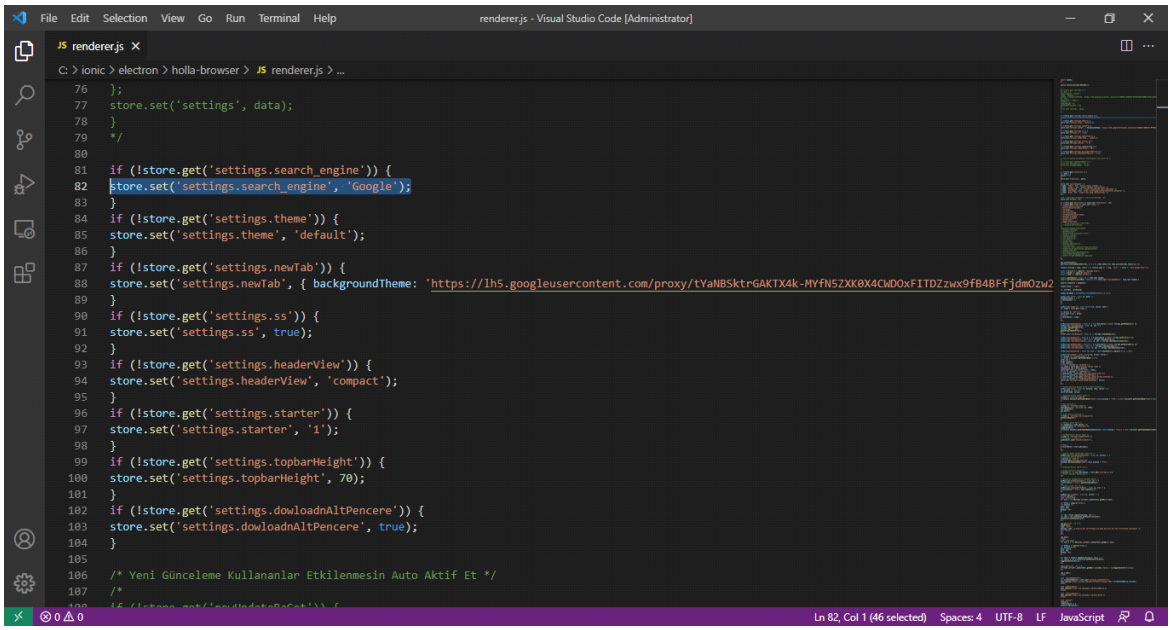


All search engines are located here.

```
store.set('searchEngines', [

{ name: 'Yaani', url: 'https://yaani.com/#q=' },

{ name: 'Google', url: 'https://google.com/search?q=' },

{ name: 'DuckDuckGo', url: 'https://duckduckgo.com/?t=yaani&q=' },

{ name: 'Startpage', url:
'https://startpage.com/do/metasearch.pl?query=' },

{ name: 'Bing', url: 'https://www.bing.com/search?q=' },

]);
```

ID; This is the search engine key name. DOMAIN_URL; Search url of the search engine

```
{ name: 'ID', url: 'DOMAIN_URL' },
```



81 lines, The part here is define search engine id available. The word 'Google' is the Default search engine key

```
if (!store.get('settings.search_engine')) {

store.set('settings.search_engine', 'Google');

}
```

Example Change

```
if (!store.get('settings.search_engine')) {
store.set('settings.search_engine', 'Example');
}

store.set('searchEngines', [
{ name: 'Yaani', url: 'https://yaani.com/#q=' },
{ name: 'Google', url: 'https://google.com/search?q=' },
{ name: 'DuckDuckGo', url: 'https://duckduckgo.com/?t=yaani&q=' },
{ name: 'Startpage', url:
'https://startpage.com/do/metasearch.pl?query=' },
{ name: 'Bing', url: 'https://www.bing.com/search?q=' },
{ name: 'Example', url: 'https://www.example.com/search?q=' }
]);
```

Components

IFOX strongly supports accessible transparency. This file gives you an insight into each of the components that make up our browser, without having to dig deep into the code to find it.

Our list uses GitHub repository links for open source projects and website links for closed source ones. License links are added to every FOSS application.

Tools

- Electron [MIT] - **In v3.5.6 and older versions**
    - Electron is the IFOX backbone. It allows us to create cross-platform desktop applications using web technologies.
    - Editors Note: While it is lacking regarding memory usage, it makes development smooth. But when optimized well, cpu or memory The problem is not seen, on the contrary, it shows very low usage and is very advantageous.
- Electron [MIT] - **In v3.5.8 and newer**
    - Electron for Content Security (ECS) is a fork of Electron created by castLabs to facilitate the use of Google's Widevine Content Decryption Module (CDM) for DRM-enabled playback within Electron, including support for Verified Media Path (VMP) and persistent license storage. It is intended to be used as a drop-in replacement for stock Electron and currently has full support for Windows and macOS platforms, with partial support for Linux (which lacks support for persistent licenses due to VMP limitations on the platform).
- Visual Studio Code [MIT]
    - Open source text editor used by IFOX's primary developer.

- GitHub Actions
    - Powerful, transparent CI / CD built directly in GitHub. All of our builds are performed using GitHub actions.
- Electron Builder [MIT]
    - A complete solution for packaging and creating the Electron, Proton Native application for MacOS, Windows and Linux, ready to deploy with "auto-update" support out of the box. It is run in GitHub Actions as shown above.

Version Control

- GitHub
    - Online Interface for Git version control software.
    - *Editors Note: We are planning to move to a self-hosted setup [Gitlab] (https://gitlab.com) or [Gitea] (https://gitea.io) to be more decentralized.*

Packets


```
"devDependencies": {
  "electron": "github:castlabs/electron-releases#v25.1.1+wvcus",
  "electron-builder": "^23.6.0",
  "electron-devtools-installer": "^3.2.0"
 },
 "dependencies": {
  "@electron/remote": "^2.0.10",
  "cors": "^2.8.5",
  "cross-fetch": "^3.0.4",
  "electron-chrome-extensions": "^3.10.1",
  "electron-context-menu": "^3.6.1",
  "electron-store": "^8.1.0",
  "electron-updater": "^5.3.0",
  "express": "^4.17.1",
  "jquery": "^3.5.1",
  "jszip": "^3.10.1",
  "npm": "^7.0.12",
```

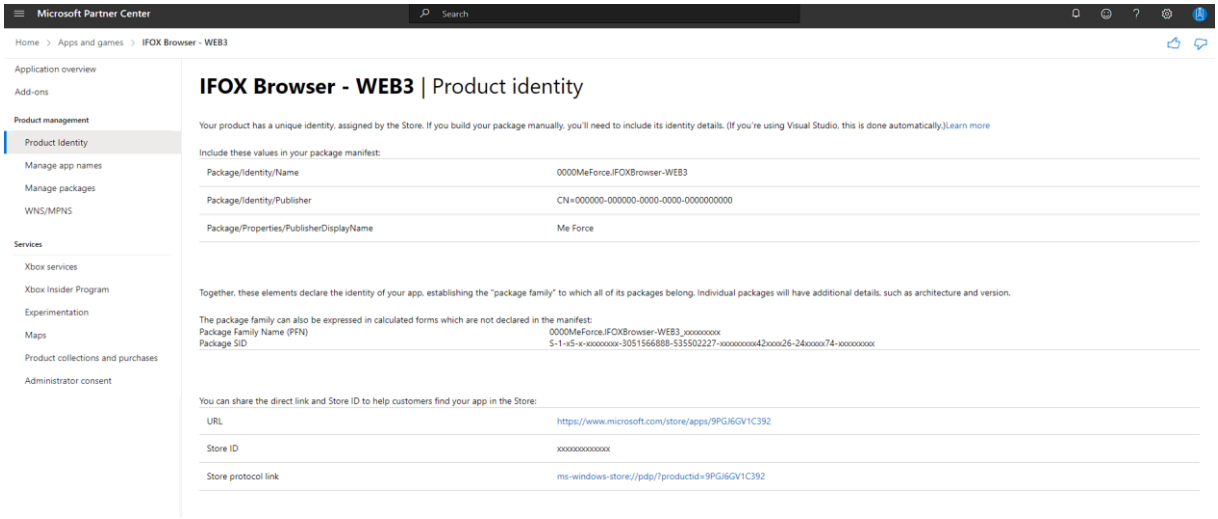
```
 "readline-sync": "^1.4.10",
 "sortablejs": "^1.10.2",
 "uuid": "^8.3.1",
 "v8-compile-cache": "^2.3.0"
}
```

Microsoft Store

https://partner.microsoft.com/en-us/dashboard/home

package.json file edit

```
 "appx": {
  "identityName": "0000MeForce.IFOXBrowser-WEB3",
  "publisher": "CN=000000-000000-0000-0000-0000000000",
  "publisherDisplayName": "Me Force",
  "applicationId": "IFOXBrowserWEB3",
  "displayName": "IFOX Browser - WEB3"
 },
```



displayName: It must be the same as the store name.
applicationId: Must be the same as displayName but no spaces and special characters

How to: https://www.codiga.io/blog/submit-electron-app-to-microsoft-store/