# Báo cáo tìm hiểu xây dựng thành phần mở rộng (extension) trên trình duyệt Chrome

*Đề bài: Xây dựng một tiện ích mở rộng trên trình duyệt chrome, khi người dùng tìm kiếm các câu trả lời trên trang web StackOverFlow tiện ích này sẽ giúp người dùng dễ dàng copy các đoạn mã nguồn bằng cách nhấn đúp chuột vào đoạn mã nguồn đó (không cần bôi đen rồi mới copy).*

## Xây dựng extension trên trình duyệt chrome

**Giới thiệu chung**

Extension (tạm dịch Thành phần mở rộng) là tổng hợp của nhiều loại file HTML, CSS, JavaScript, ảnh, những file được sử dụng trên nền tảng web được gộp lại với nhau, giúp cho trải nghiệm khi sử dụng Google Chrome được tốt hơn. Những Extension được xây dựng sử dụng công nghệ web và có thể sử dụng chung những APIs do trình duyệt cung cấp tới web mở.

Extensions có nhiều chức năng. Chúng có thể chỉnh sửa nội dung website mà người dung nhìn thấy và tương tác được và cũng có thể thay đổi hành vi của chính trình duyệt.

Chúng làm cho trình duyệt mang tính cá nhân hóa hơn.

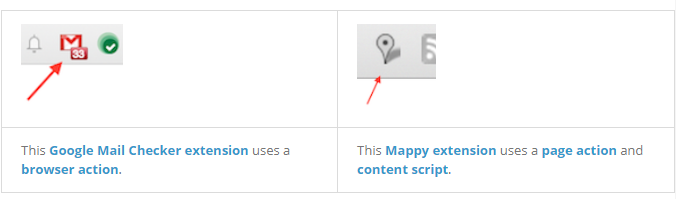
**Những File Mở rộng**

Extension có nhiều loại file trong nó và nhiều thư mục, nhưng chúng đều có chung đặc điểm là yêu cầu file manifest. Một extension đơn giản nhất bao gồm một file manifest đi kèm với icon toolbar của nó.

Những file manifest có tên manifest.json cung cấp cho trình duyệt thông tin về extension, như là những file quan trọng nhất và extension sẽ được sử dụng vào việc gì.



Những extension phải bao gồm một icon nằm trên phần toolbar của trình duyệt. Icon này cho phép người dùng dễ dàng biết rằng extension được cài đặt và cũng dễ dàng tương tác và sử dụng. Phần lớn người dùng tương tác với những extension này bằng việc sử dụng một cửa sổ popup khi họ click vào icon trên.



**Referring to files**

Một file trong gói extension phải được khai báo sử dụng relative URL (đường dẫn tương đối), được coi là những file trong trang HTML bình thường.



Ngoài ra, mỗi file có thể được truy cập bằng cách sử dụng đường dẫn URL tuyệt đối.



Với đường dẫn tuyệt đối, <*extensionID*> là một tên định danh duy nhất mà hệ thống extension tạo ra cho riêng mỗi extension. Những ID tương ứng với những extension được xem bằng cách truy cập vào đường dẫn URL *chrome://extensions*. <*pathToFile*> là địa chỉ của file trong thư mục extension, nó được biểu diễn bằng địa chỉ tương đối (có gốc là địa chỉ của top folder).

Khi làm việc trên một extension được unpack, ID của extension đó có thể thay đổi. Đặc biệt, id của extension bị unpack sẽ thay đổi nếu extension được load từ một đường dẫn khác. ID cũng sẽ thay đổi khi extension được package (đóng gói). Nếu code của extension dựa vào một URL tuyệt đối, nó có thể sử dụng *chrome.runtime.getURL()* để tránh mã hóa cứng một id trong khi dev.

**Cấu trúc file của một Extension**

Nhiều file extension đều bao gồm một số những thành phần sau:

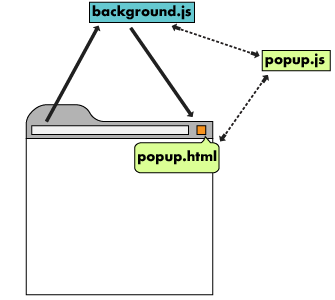
***Manifest***

***Background Script***

* Là một trình điều khiển sự kiện của extension, chứa các bộ lắng nghe sự kiện của trình duyệt, là một chức năng quan trọng của extension. Nó sẽ nằm im cho đến khi một sự kiện được kích hoạt, sau đó thực thi một tập lệnh logic nào đó. Một mã script trong nền có hiệu lực chỉ khi được load khi cần và không được load khi nó rảnh rỗi.

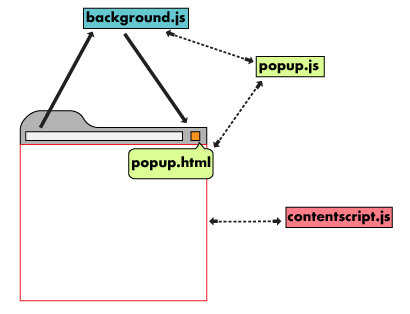
***UI Elements***

* Một giao diện người dùng cần hướng mục đích và nhỏ nhất có thể. UI được tùy chỉnh hoặc phát triển dựa vào trình duyệt và tập trung vào nó. Phần lớp extension có một browse action và một page action, nhưng cũng có thể chứa các form UI khác, như context menus sử dụng omnibox hoặc tạo một keyboard shortcut.
* Trang UI của extension như popup có chứa những trang HTML gốc và mã JavaScript logic. Những Extension cũng có thể gọi *tabs.create* hoặc *window.open()* để hiển thị dạng file HTML truyền thống được biểu diễn trong extension.
* Một extension sử dụng một page action và popup có thể sử dụng declarative content PI (tạm dịch API khai báo nội dung) để thiết lập luật trong những đoạn script nền khi popup khả dụng với người dùng. Khi điều kiện được thỏa mãn, đoạn script dưới nền này sẽ giao tiếp với popup để khiến icon của nó cho phép người dùng click vào.

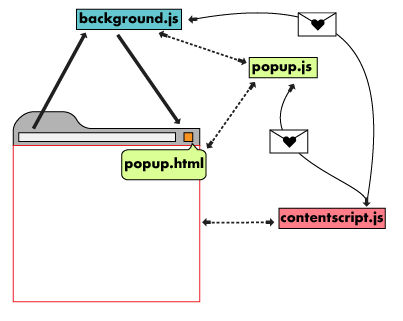


***Content Script***

Các tiện ích mở rộng có thể đọc hoặc ghi vào các trang web sử dụng content script. Tập lệnh chứa mã thực thi bằng JavaScript trong ngữ cảnh của một trang được tải lên trình duyệt. Tập lệnh này đọc và sửa đổi kiến trúc DOM của các trang web mà trình duyệt truy cập.



Những tập lệnh (content script) có thể giao tiếp với các tiện ích cha của nó bằng việc trao đổi thông điệp và lưu trữ giá trị bằng storage API.



***Options Page***

Giống như các tiện ích mở rộng cho phép người dùng tùy chỉnh trình duyệt Chrome, trang Option Page cho phép tùy chỉnh các tiện ích mở rộng. Tùy chọn có thể sử dụng để kích hoạt các tính năng và cho phép người dùng chọn chức năng nào phù hợp với nhu cầu của họ.

## Thiết kế và xây dựng thành phần extension:

### Thiết kế

**Extension là một thư mục bao gồm:**

* Một file manifest.json: là file bắt buộc phải có của mỗi extension, nó chứa thông tin chi tiết về extension như tên, phiên bản, các ảnh được sử dụng, quyền sử dụng trong chrome.
* một file icon: chứa icon sẽ hiển thị trên thanh toolbar của trình duyệt.
* một file inject.js: chứa đoạn code js xử lý sự kiện double click của chuột đồng thời lưu vùng code trong stackoverflow vào clipboard để sử dụng sau. Đóng vai trò như contentscript.js
* Không sử dụng cửa sổ popup.
* Về mô hình thiết kế giống với mô tả bên trên ở phần cấu trúc file Extension.

**Kịch bản:**

* Khi người dùng double click nút chuột trái vào một vùng chứa code (dạng text) trên trang web stackoverflow thì ngay lập tức sẽ đọc đoạn code trong vùng đó ghi vào clipboard.
* Khi double click nút chuột trái vào các vùng khác thì vẫn hoạt động như bình thường như khi không có extension.

### Xây dựng

1. **Manifest.json**

Các trường được sử dụng bao gồm:

* “name”: “CpCode\_SO” – Tên extension hiển thị trong phần quản lý
* “version”: “1.0.1” – Phiên bản hiện tại là 1.0.1
* “manifest\_version”: 2 – Phiên bản của manifest, thông thường là 2
* “description”, “author” – mô tả và tác giả của extension
* “permission”: gồm các quyền của extension sẽ được thực hiện trên chrome, ở đây cần 2 quyền là ghi vào clipboard (clipboardWrite) và quyền storage (quyền lưu trữ)
* “browser\_action”: trường con “default\_icon” để cài đặt icon mặc định cho extension sẽ được hiển thị trên toolbar



* “content\_scripts”: khai báo đường dẫn mã js xử lý cho extension, giao thức http, https, và vị trí khởi chạy.
* “offline\_enabled”: trường này cung cấp khả năng hoạt động ngay cả khi không kết nối mạng

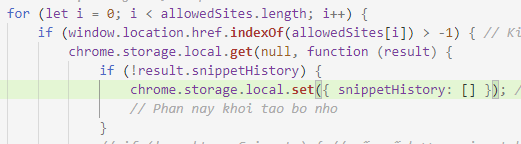
1. **Inject.js**

* File này chứa mã nguồn xử lý công việc inject sự kiện double click của người dùng:
* Các đoạn code chính:

+ Danh sách các trang web cho phép



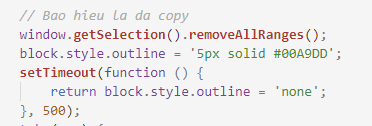
+ Khởi tạo vùng nhớ để lưu code



+ Khởi tạo đối tượng mảng lưu từng block code một. Bắt sự kiện double click chuột trái, lấy block code tương ứng, lưu vào đối tượng Range phạm vi toàn trình duyệt (window), sau đó chạy lệnh ‘copy’ để lưu đoạn code vào clipboard.



+ Sau khi đã lưu vào clipboard thì gọi hàm hủy đối tượng Range, đồng thời báo hiệu là đã copy vào clipboard. Ở đây khi copy thành công sẽ xuất hiện một đường bao quanh khối code trên giao diện web,có màu xanh và xuất hiện trong 500ms.

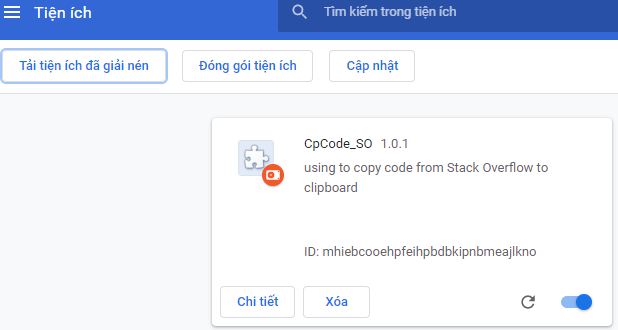


## Kết quả kiểm thử extension

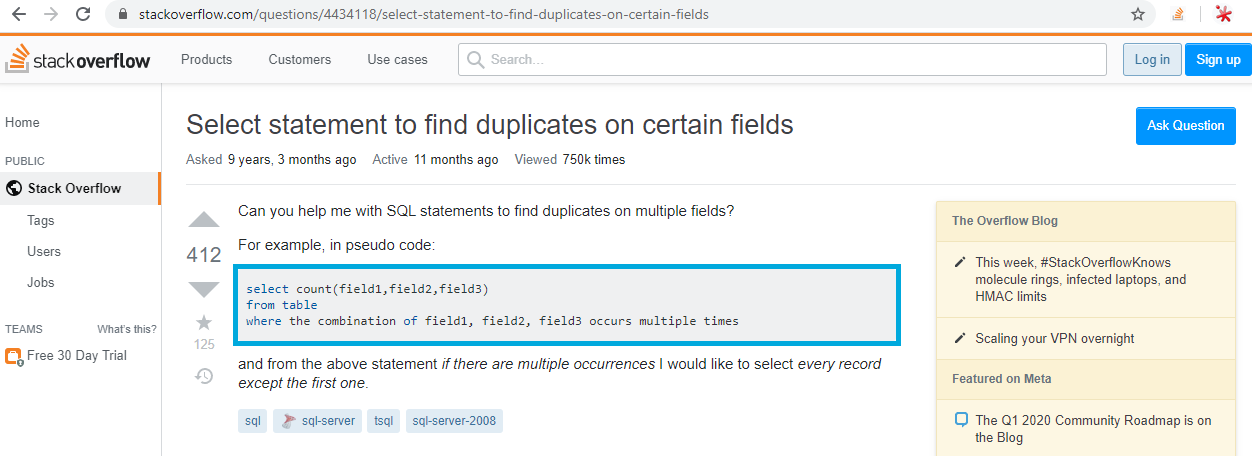
* Cài đặt bằng cách vào phần tiện ích mở rộng trên trình duyệt, bật chế độ dành cho nhà phát triển, chọn Tải tiện ích đã giải nén và trỏ đường dẫn tới thư mục chứa mã nguồn.
* Sau khi cài đặt thì trên toolbar của trình duyệt sẽ có biểu tượng icon:



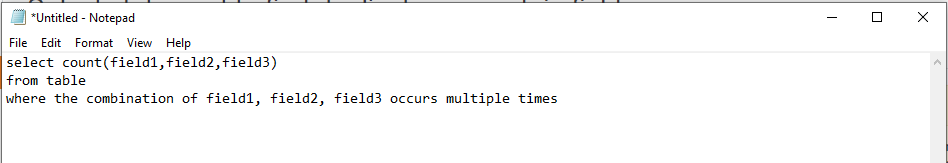
* Trong phần quản lý tiện ích mở rộng của trình duyệt sẽ hiển thị:



* Khi double click nút trái chuột vào một ô chứa code trên trang web stackoverflow.com thì sẽ hiển thị viền màu xanh bao quanh ô code báo là đã copy đoạn code vào trong clipboard



* Khi Ctrl+V vào một vùng edit nào đó thì nó sẽ hiển thị ra đoạn code đã copy: ví dụ ở đây là notepad:



## Tài liệu tham khảo

* https://developer.chrome.com/extensions
* https://www.w3schools.com/jsref