**BÁO CÁO MÔN PHÁT TRIỂN HỆ THỐNG WEB AN TOÀN**

**TÌM HIỂU CHUNG VỀ PHẦN MỞ RỘNG CỦA TRÌNH DUYỆT CHROME**

**VÀ**

**XÂY DỰNG MỘT EXTENSION THEO YÊU CẦU**

**Giảng viên hướng dẫn** : *Nguyễn Mạnh Tuấn*

**Sinh viên** : *Nguyễn Thanh Sơn*

**Mã số sinh viên**: *20163546*

**Trường** : *Đại học Bách khoa Hà Nội*

**Email** : [*son.nt163546@sis.hust.edu.vn*](mailto:son.nt163546@sis.hust.edu.vn)

**Phone** : *0358228333*

MỤC LỤC

1. TÌM HIỂU CHUNG VỀ PHẦN MỞ RỘNG CỦA TRÌNH DUYỆT CHROME (EXTENSION)
2. Extension là gì?
3. Cấu trúc của một extension

2.1. Tổng quan

* 1. Kiến trúc

1. Tải một extension thủ công của bạn lên trình duyệt để kiểm thử
2. XÂY DỰNG MỘT ỨNG DỤNG EXTENSION
3. KẾT LUẬN
4. TÌM HIỂU CHUNG VỀ PHẦN MỞ RỘNG CỦA TRÌNH DUYỆT CHROME (EXTENSION)
5. **Extension là gì ?**

Extension là một chương trình phần mềm hỗ trợ trải nghiệm duyệt web. Chúng cho phép người dung điều chỉnh chức năng của hành vi của Chrome theo nhu cầu hoặc sở thích cá nhân. Chúng được xây dựng trên những công nghệ web cơ bản đó là HTML, JavaScript và CSS. Mỗi thành phần của extension phụ thuộc vào chức năng của nó và không nhất thiết yêu cầu mọi cài đặt được thiết lập.

1. **Cấu trúc của một extension**
   1. **Tổng quan**

Các extension được xây dựng từ các thành phần gồm HTML, CSS, Javascript, các file ảnh và các loại file khác dựa trên nền tảng web, cải thiện trải nghiệm duyệt web của Google Chrome đối với người dùng. Extension được xây dựng sử dụng công nghệ web và có thể sử dụng cùng các API mà trình duyệt cung cấp cho web mở.

Các extension có thể thực hiện được nhiều chức năng khác nhau. Chúng có thể chỉnh sửa nội dung web mà người dùng thấy và tương tác hoặc mở rộng và thay đổi hành vi của trình duyệt.

Extensions có thể khác nhau ở loại file và số lượng các thư mục, nhưng chúng bắt buộc phải có một file ***manifest***. Một số tiện ích cơ bản nhưng hữu dụng có thể chỉ bao gồm tập manifest và biểu tượng trên thanh công cụ của nó.

File ***manifest.json*** , gửi đến trình duyệt các thông tin , cấu hình của extension, ví dụ như thông tin các file quan trọng nhất và các khả năng extension có thể sử dụng.

{

"name": "My Extension",

"version": "2.1",

"description": "Gets information from Google.",

"icons": {

"128": "icon\_16.png",

"128": "icon\_32.png",

"128": "icon\_48.png",

"128": "icon\_128.png"

},

"background": {

"persistent": false,

"scripts": ["background\_script.js"]

},

"permissions": ["https://\*.google.com/", "activeTab"],

"browser\_action": {

"default\_icon": "icon\_16.png",

"default\_popup": "popup.html"

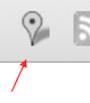
}

}

Extension phải có một biểu tượng nằm trên thanh công cụ của trình duyệt. Các biểu tượng trên thanh công cụ cho phép truy cập dễ dàng và giúp người dùng biết extension nào đã được cài đặt. Hầu hết người dùng sẽ tương tác với extension thông qua một **popup** bằng cách click vào biểu tượng.



**Google Mail Checker extension** sử dụng **browser action**



**Mappy extension** sử dụng page action và **content script**

Những file của một extension có thể được tham chiếu bằng cách sử dụng URL, giống như HTML.

<img src="images/my\_image.png">

Ngoài ra, các file còn có thể truy cập trực tiếp bằng đường dẫn tuyệt đối

.

**chrome-extension://**<extensionID>**/**<pathToFile>

Trong đường dẫn tuyệt đối, ID của extension là duy nhất mà hệ thống sinh ra cho mỗi extension. Những ID cho tất cả extension được tải có thể nhìn thấy thông qua URL **chrome://extensions** . Đường dẫn đến vị trí của file. Phần <pathToFile> là vị trí của file dưới các thư mục cao nhất.

* 1. **Kiến trúc**

Kiến trúc của một tiện ích sẽ phụ thuộc vào chức năng của chúng., nhưng các extension phức tạp sẽ bao gồm nhiều thành phần:

* **Manifest**
* **Background Script**
* **UI Elements**
* **Content Script**
* **Options Page**

**Background Script**

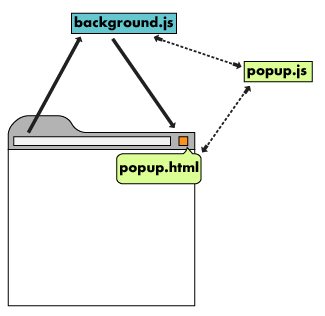
Background script là trình xử lí sự kiện của extension; nó bao gồm bộ lắng nghe các sự kiện của trình duyệt mà nó quan trọng đối với extension. Nó nằm im cho đến khi có một sự kiện được kích hoạt sau đó thực hiện các chỉ thị logic. Một background script hiệu quả chỉ được tải khi cần thiết và thoát khi nhàn rỗi.

**UI Elements**

Một giao diện người dùng của extension nên có mục đích và tối thiểu. UI nên cải thiện hoặc nâng cao trải nghiệm duyệt web mà không bị mất tập trung bởi nó. Hầu hết các extension có một **browser action** hoặc **page action**, nhưng có thể bao gồm các form khác của UI, ví dụ như **context menu (menu xuất hiện khi nhấp chuột phải),** hoặc ominibox, hoặc tạo một **keyboard shortcut.**

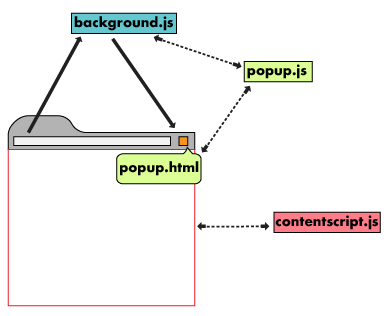
Trang giao diện extension, ví dụ như **popup,** có thể bao gồm trang HTML thông thường với Javascript logic. Các extension cũng có thể gọi **tabs.create** hoặc *window.open()* để hiển thị các file HTML bổ sung có trong extension.

Một extension sử dụng một page action và một popup có thể sử dụng declare content API để thiết lập luật trong background script cho khi popup sẵn sàng đối với người dùng. Khi gặp các điều kiện cần, background script giao tiếp với popup để khiến biểu tượng có thể click vào đối với người dùng.

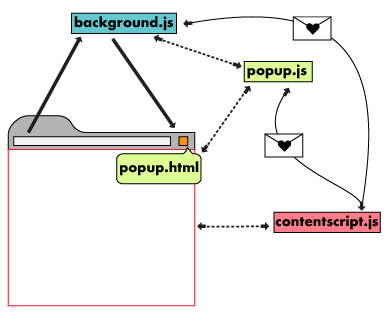


**Content script**

Extensions tương tác trực tiếp tới trang web thông qua file **content script**. Content script bao gồm **javascript** mà kích hoạt trong ngữ cảnh của trang web, được tải vào trong trang web. Content script đọc và chỉnh sửa các DOM ( Document Objetc Model ) của trang web mà trình duyệt đang duyệt.



Content script có thể giao tiếp vói với các thành phần các của extension bằng các ***messages*** và lưu trữ giá trị trong các ***storage*** API.



**Options Page**

Trang web tùy chọn cho phép người dùng tùy chỉnh trình duyệt Chrome, **options page** cho phép sử cải thiện của extension. Các tùy chọn có thể được sử dụng để cho phép các chức năng và cho phép người dùng chọn chức năng cần thiết.

**Sử dụng Chrome APIs**

Ngoài việc truy cập vào cùng APIs như trang web, các extension cũng có thể sử dụng các APIs dành riêng cho extension để tích hợp chặt chẽ vào trình duyệt. Ví dụ như các tiện ích có thể và trang web có thể cả truy cập vào phương thức chuẩn ***window.open()*** để mở mộ URL , nhưng các extension có thể chỉ định cửa sổ nào mà URL nên được hiển thị bằng cách sử dụng Chrome API **tabs.create** thay thế.

1. **Tải một extension thủ công của bạn lên trình duyệt để kiểm thử**

Tải một extension thủ công ( không thuộc Web Store ) lên trình duyệt

để phục vụ tiện ích cá nhân hoặc để kiểm thử phần mềm extension, ta có thể làm như sau :

1. Truy cập url : ***chrome://extensions*** hoặc vào Menu -> Tools -> Extensions
2. Bật chế độ nhà phát triển
3. Lựa chọn “Tải tiện ích đã nén”
4. Chọn folder chứa extension của bạn

Như vậy , extension đã được tải lên trình duyệt và bạn có thể kiểm thử hoặc sử dụng

1. **XÂY DỰNG MỘT ỨNG DỤNG EXTENSION**

Yêu cầu : ***Xây dựng một tiện ích mở rộng trên trình duyệt chrome, khi người dùng tìm kiếm các câu trả lời trên trang web Stack Overflow 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)***

Đầu tiên, chúng ta bắt đầu bằng file **manifest.json :**

{

    "name": "Copy Code Stackoverflow",

    "version": "1.0",

    "description": "This is the tool for copying code on stackoverflow web pages!",

    "manifest\_version": 2

}

Khai báo các trường cơ bản của extension, bao gồm tên extension, phiên bản extension, mô tả extension và phiên bản manifest (giá trị yêu cầu là 2 kể từ chrome 18).

Ở extension này, em lựa chọn lập trình không cần sử dụng đến file background.js vì muốn tăng hiệu suất cho ứng dụng. Bên cạnh đó, theo yêu cầu của extension , ta không cần phải xử lí các sự kiện của trình duyệt để phục vụ các tác vụ của extension, do đó ta có thể bỏ qua sử dụng backgound script.

Tiếp theo, ta sẽ thiết kế cửa sổ popup hiện ra khi nhấp chuột (click) vào biểu tượng icon trên thanh công cụ trình duyệt.

**popup.html**

<!DOCTYPE html>

<html>

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<head>

    <style>

    </style>

</head>

<body>

    <div class="alert alert-danger">Current page :</div>

    <div class="js-currentURL alert alert-light"></div>

    <div class=" text-center alert-primary">DOUBLE-CLICK TO PROGARMING CODE TO COPY</div>

    <h7 class="text-center font-weight-bold" >Copied code</h7>

    <div id="js-copiedText" class= "ml-3"></div>

    <div class="alert alert-success mt-2">Author:  <a href="https://www.facebook.com/thanhson7198" class="text-success" ><span class="font-weight-bold">  Thanh Son Nguyen</span> </a> </div>

</body>

<script src="popup.js"></script>

<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>

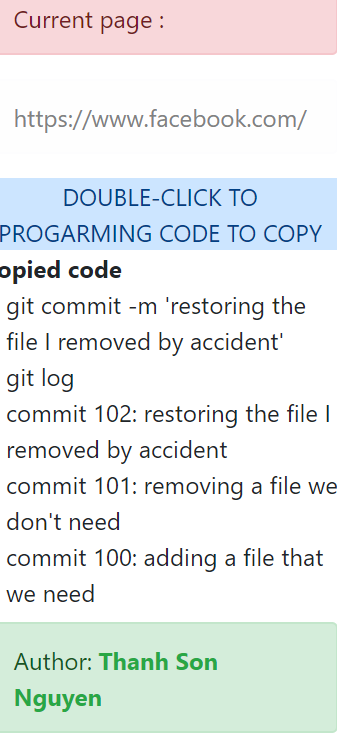
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>

</html>

Nội dung của popup nằm trong thẻ body (tương tự một trang web ) . Ở đây, em thiết kế thành 3 phần trong popup để hiển thị 3 thông tin khác nhau bao gồm :

* *URL hiện tại ( đường dẫn đến trang web câu hỏi trên stackoverflow mà người dùng đang tương tác)*
* *Nội dung đoạn text được copy*
* *Tên tác giả ( đường dẫn đến facebook)*

Nội dung của file popup sau khi tải extension và render :



File popup.js xử lí các sự kiện do người dùng tương tác với popup, được thêm vào file popup.html thông qua thẻ <script src=”popup.js”></script>

File popup.js không chỉ xử lí các tương tác của người dùng, mà còn nhận dữ liệu từ phía file content.js và hiển thị ra ngoài popup.

chrome.tabs.query({ 'active': true, 'lastFocusedWindow': true }, function (tabs) {

    var url = tabs[0].url;

    let currentURL = document.querySelector(".js-currentURL");

    currentURL.innerText = url;

    chrome.tabs.query({active: true, currentWindow: true}, function(tabs) {

        var copiedText = "";

        chrome.storage.local.get('copiedText', function (result) { // lấy đoạn code được copy lưu tạm

                                                                    // lưu tạm trong storage

            copiedText = result.copiedText;

            // hiển thị đoạn code được copy ra ngoài popup

            document.querySelector("#js-copiedText").innerText = copiedText;

        });

      });

});

File trên sử dụng các API **Chrome.storage** để lấy dữ liệu được lưu trên trong **storage,**  cụ thể dữ liệu được lưu là đoạn code được copy từ trình duyệt web, được lưu bên phía file **content.js**

File content.js xử tương tác của người dùng với trang web. Ở đây, em xử lí thao tác double-click chuột của người dùng vào đoạn code tương ứng. Đầu tiên, sau khi double-click , đoạn code tương ứng sẽ được chọn toàn bộ (bôi đen toàn bộ ) thay vì chỉ một phần nhỏ của code. Tiếp theo, em viết lại thao tác copy của của trình duyệt bằng javascript : lưu toàn bộ đoạn code được bôi đen vào clipboard, sau đó đọc ra từ clipboard và lưu vào **storage** bằng API **Chrome.storage** để phía popup.js lấy ra hiển thị ở popup.

let codeTagsHC = document.getElementsByTagName("code");

let codeTags = [].slice.call(codeTagsHC);

// hàm chọn (bôi đen) toàn bộ code được double-click

function selectText(node) {

    if (document.body.createTextRange) {

        const range = document.body.createTextRange();

        range.moveToElementText(node);

        range.select();

    } else if (window.getSelection) {

        const selection = window.getSelection();

        const range = document.createRange();

        range.selectNodeContents(node);

        selection.removeAllRanges();

        selection.addRange(range);

    } else {

        console.warn("Could not select text in node: Unsupported browser.");

    }

}

// theo cấu trúc trang stackoverflow, tất cả đoạn code được đặt trong thẻ <code>

// việc cần làm là lấy toàn bộ code bên trong thẻ code và bôi đen toàn bộ đoạn code đó

// xử lí lại theo tác copy khi lưu vào clipboard, và lưu nội dung đoạn code được copy vào storage

for (let tag of codeTags) {

    tag.addEventListener("dblclick", function () {

        console.log(tag);

        selectText(tag); // bôi đen toàn bộ code

        navigator.clipboard.writeText(tag.innerText).then(function () { // ghi code vào clipboard

            console.log("clipboard successfully set");

            navigator.clipboard.readText().then(function (clipText) { // đọc đoạn code từ clipboard

                console.log(clipText);

                chrome.storage.local.set({'copiedText': clipText}); // sau đó lưu vào trong storage

            });

        }, function () {

            console.log("clipboard write failed");

        });

    });

}

Cập nhật file **manifest.json** ,

{

    "name": "Copy Code Stackoverflow",

    "version": "1.0",

    "description": "This is the tool for copying code on stackoverflow web pages!",

    "permissions": [

        "tabs",

        "activeTab",

        "declarativeContent",

        "storage"

    ],

    "content\_scripts": [

        {

            "matches": ["https://stackoverflow.com/questions/\*"],

            "js": ["copy.js"]

        }

    ],

    "browser\_action": {

        "default\_popup": "popup.html",

        "default\_title": "Copying programing code on stackoverflow web pages tool",

        "default\_icon": {

            "16": "images/get\_started16.png",

            "32": "images/get\_started32.png",

            "48": "images/get\_started48.png",

            "128": "images/get\_started128.png"

        }

    },

    "icons": {

        "16": "images/get\_started16.png",

        "32": "images/get\_started32.png",

        "48": "images/get\_started48.png",

        "128": "images/get\_started128.png"

    },

    "manifest\_version": 2

}

“**content\_script**”. “**matches**” định nghĩa trang url mà extension có hiệu lực, mà cụ thể ở đây là url [*https://stackoverflow.com/questions/\**](https://stackoverflow.com/questions/*). Điều này có nghĩ là extension chỉ có hiệu lực tại trang hỏi đáp trên stackoverflow, các trang web khác , hay thậm chí các url khác của stackoverflow , extension không có hiệu lực.

**“content\_script” . “js”** chỉ định file javascript tương tác với nội dung trang web, cụ thể là file **content.js**

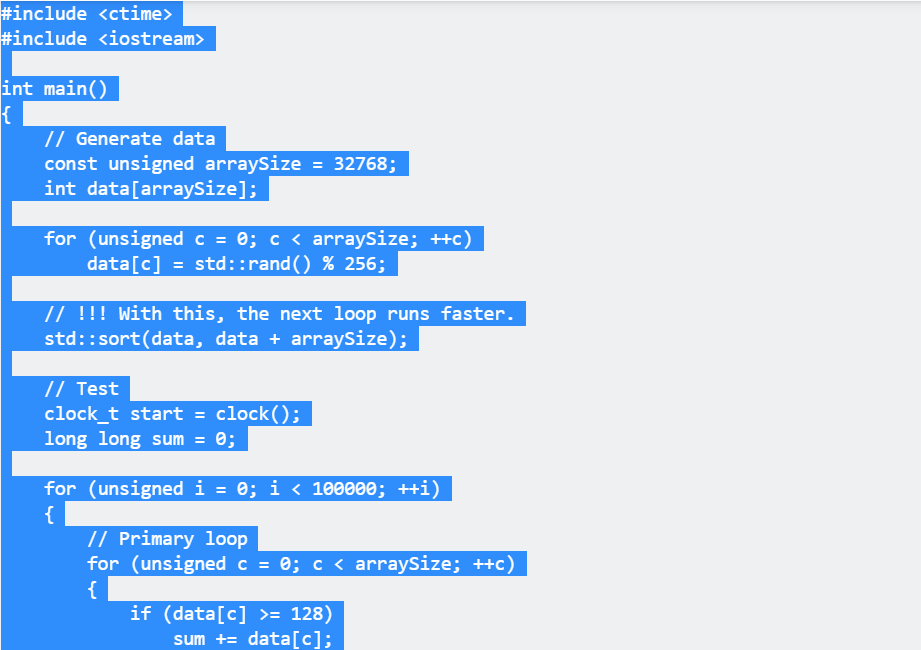
**“browser\_action”** định nghĩa trang html của popup hiện ra khi click vào icon và title hiện ra khi rê (hover) chuột qua icon, cụ thể là file **popup.html**. “**defautl\_icons**” chỉ định đường dẫn đến file hình ảnh icon khi hiển thị trên thanh toolbar tùy kích thước”

“**icons**” chỉ định đường dẫn hiển thị hình ảnh icon của extension tại đường kho extension theo đường dẫn **chrome://extensions**

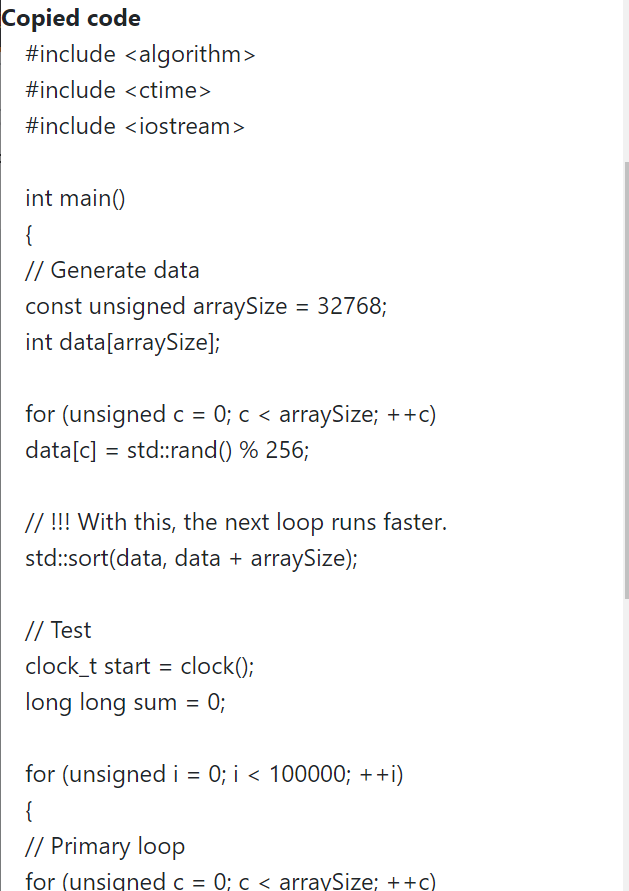
“**permissions**” là nơi khai báo các chrome.\* APIs mà extension sử dụng. “Permission” giúp hạn chế thiệt hại nếu tiện ích mở rộng hoặc ứng dụng của bạn bị phần mềm độc hại xâm phạm.

Sau đây là cách extension hoạt động tại url [***https://stackoverflow.com/questions/\****](https://stackoverflow.com/questions/*)

Double-click vào đoạn code, toàn bộ đoạn code được chọn sẽ được bôi đen



Đoạn code được bôi đen sẽ tự động được copy vào clipboard , vào hiển thị trên **popup**



1. KẾT LUẬN

Extension bên trên là một tiện ích mở rộng đơn giản, thực hiện những thao tác đơn giản. Thực tế , những nhà phát triển đã sáng tạo ra hàng ngàn extension khác nhau , trong đó có những extension thực hiện những công việc cực kì hữu ích với khối lượng thao tác không hề nhỏ. Lợi ích của các extension khiến chúng trở thành một phần quan trọng của ứng dụng chrome, giúp người dùng cải thiện trải nghiệm duyệt web cũng như các công việc khác.