

Development of Facebook Bookmark with Chrome Extension



2014년 5월 9일

작성자 : 이동엽(dongyub19), 전성윤(roland)

Email : dongyu63@gmail.com, roland.korea@gmail.com

<http://newheart.kr>

Contents

Chapter 1 Overview

Chapter 2 Chrome Extension

2.1 What is Chrome Extension

2.2 How to begin development with Chrome Extension

Chapter 3 Development of Facebook Bookmark

3.1 Exploring Our Application

3.2 Techniques in Our Application

Chapter 4 Future plan of Developing Facebook Bookmark

Chapter 5 References

Chapter 1 Overview

Usually, when I use Facebook, I need some function such as bookmark or something to remember the url so that I can remember specific information; such as a part of an article, videos, and so on, from the url. However, Facebook doesn't offer anything like bookmark function, so I decide to develop bookmark function. I choose Google Chrome Extension; because, the number of Chrome users is growing, as a development tool in developing Facebook bookmark function. In this document, I'll describe Chrome Extension and how I develop Facebook bookmark function with Chrome Extension. You can use our application via the following link.

<https://chrome.google.com/webstore/detail/bookmark-on-facebook/khdcaedcacekacbgackgpkbmkcfie>

Chapter 2 Chrome Extension

2.1 What is Chrome Extension

Chrome Extension is a software that you can modify and enhance functionality of the Chrome browser. You can easily develop applications in Chrome Extension using web languages such as HTML, JavaScript, and, CSS.

2.2 How to begin development with Chrome Extension

As a first step, you need to create a manifest file named manifest.json. In this case, I'll show you our project's manifest.json file with picture(1).

```
{
  "content_security_policy": "script-src 'self' https://ajax.googleapis.com; object-src 'self'",
  "description": "You can manage the bookmarks of facebook!",
  "icons": {
    "48": "img/48.png",
    "16": "img/16.png"
  },
  "browser_action": {
    "default_icon": {
      "19": "img/19.png"
    },
    "default_title": "Bookmark On Facebook",
    "default_popup": "popup.html"
  },
  "background": {
    "scripts": [
      "background.js"
    ]
  },
  "manifest_version": 2,
  "name": "Bookmark on Facebook",
  "offline_enabled": true,
  "permissions": [
    "bookmarks"
  ],
  "web_accessible_resources": [
    "jquery/js/jquery.min.map",
    "jquery/css/*"
  ],
  "version": "0.0.0.1",
  "content_scripts": [{
    "matches": [ "*//*.facebook.com/*" ],
    "js": [
      "jquery/js/jquery-1.10.2.min.js",
      "bookmark.js",
      "jquery/js/jquery-ui-1.10.4.min.js"
    ]
  }],
  "run_at": "document_end"
}]
}
```

Picture(1) manifest.json

As you can see manifest.json file, this is just JSON-formatted table of contents. It contains properties such as extension's name, description, its version number, and so on. In order to display your application, you need to set up browser_action part to create browser action. To declare to Chrome what the extension is going to do and what permissions are needed to do specific job, you need to write manifest.json file.

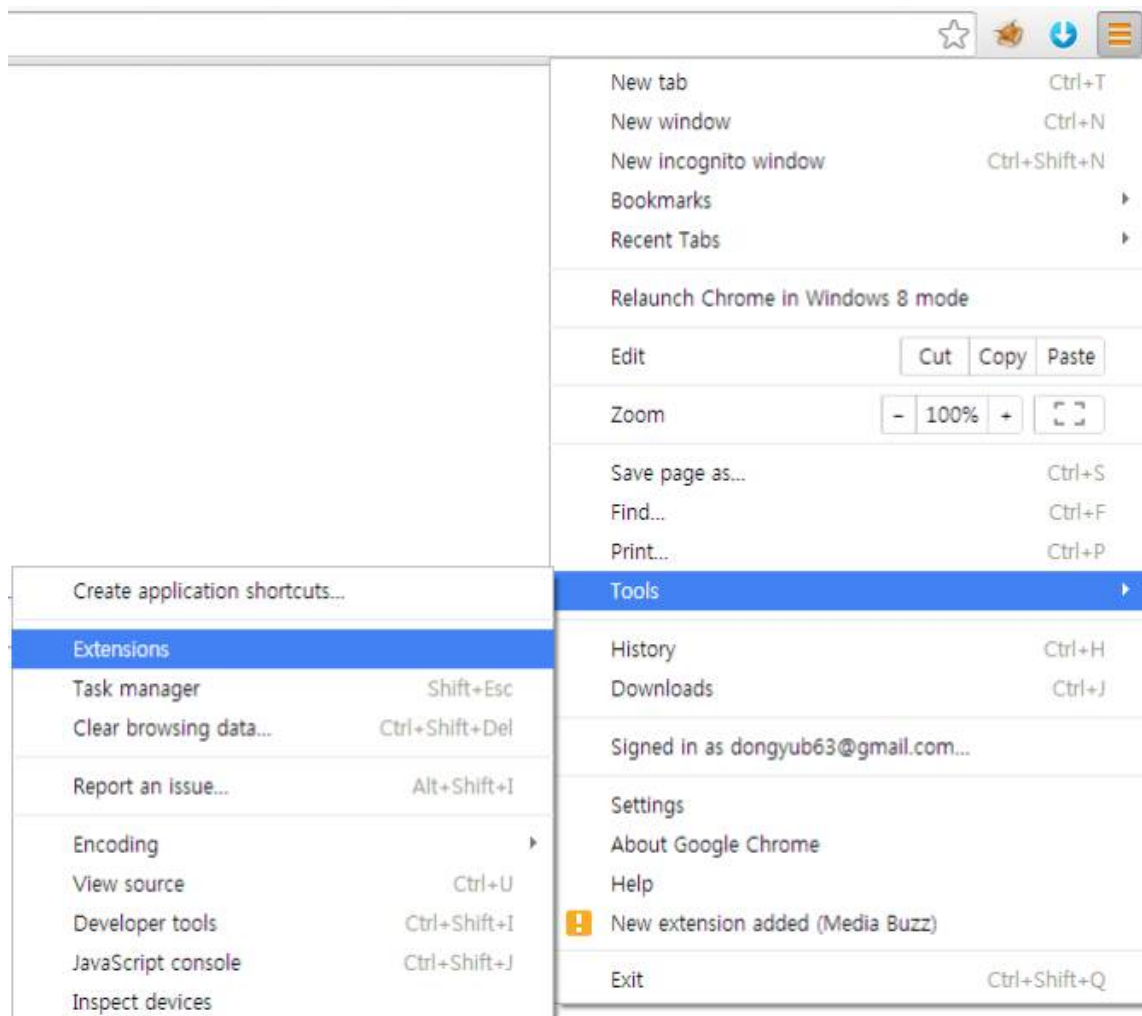
There are lots of things that you need to set up so that I'll be able to explain few things you should declare. First, you should declare manifest_version, extension's name, description, and, version. Its name will be used in Web Store to display your extension to users. The last things you should declare are browser_action and permissions.

After you write manifest.json file, you should move resources that you've declared at manifest.json file to same directory with manifest.json as seen below picture(2).

img	2014-05-08 오전...	파일 폴더	
jquery	2014-05-10 오후...	파일 폴더	
background.js	2014-05-10 오전...	JavaScript 파일	6KB
bookmark.js	2014-05-10 오후...	JavaScript 파일	9KB
manifest.json	2014-05-08 오후...	JSON 파일	1KB
popup.css	2014-05-10 오후...	CSS 스타일시트 ...	1KB
popup.html	2014-05-08 오후...	Chrome HTML D...	2KB
popup.js	2014-05-08 오후...	JavaScript 파일	4KB

Picture(2) Reside resources to same directory

After finishing the steps that are explained above, you should load your directory in <chrome://extensions/> page. I'll explain how to load your directory as seen below pictures.



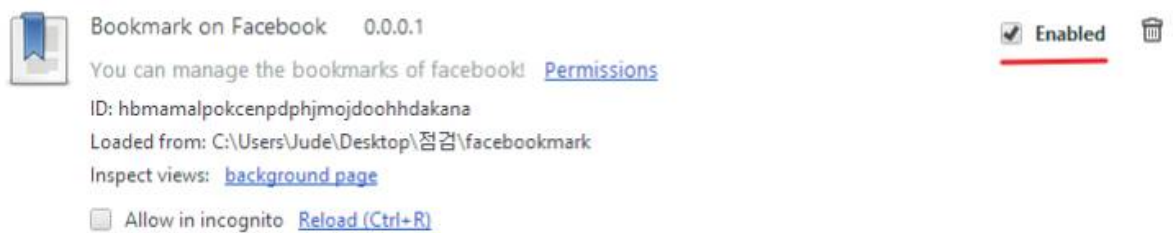
Picture(3) Go to chrome://extensions/ page



Picture(4) Select 'Load unpacked extension...' button



Picture(5) Select your directory



Picture(6) Set your application's status as enabled

Chapter 3 Development of Facebook Bookmark

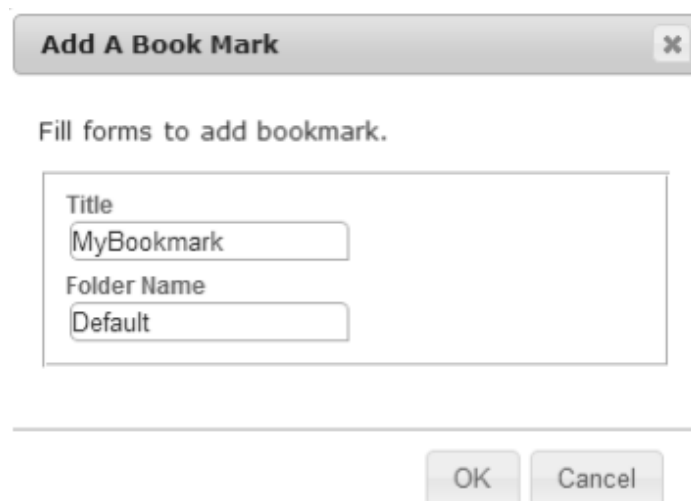
3.1 Exploring My Application

From now on, I will explain briefly about how to use our application and some techniques that we use.



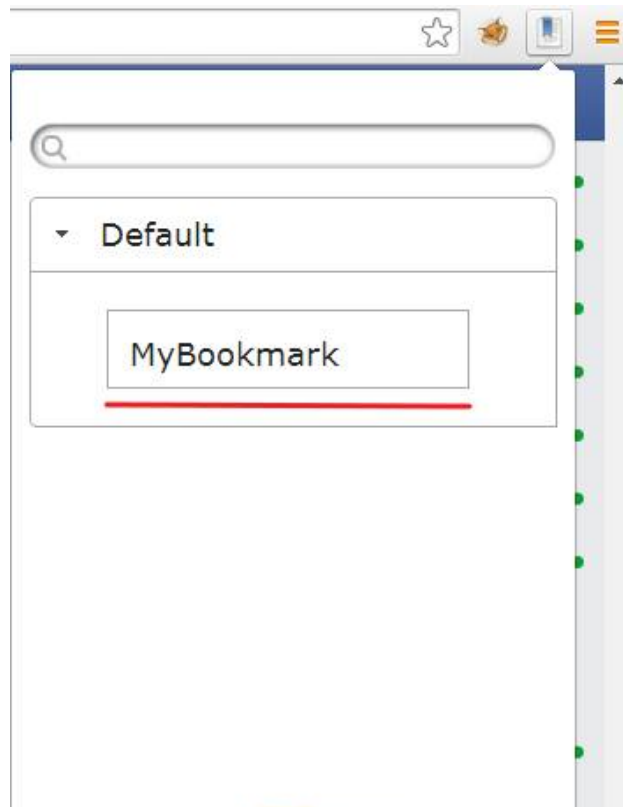
Picture(7) Bookmark buttons underlined

If you click the star button in Picture(7), you can add a bookmark as seen below picture(8).



Picture(8) Adding a bookmark in dialog

After you finish steps above, you can see the bookmark list that you just add as seen below picture (9).



Picture(9) List of bookmark through application

If you click any of the bookmarks in the list, Chrome will create a new tab with the added url as seen below picture(10).

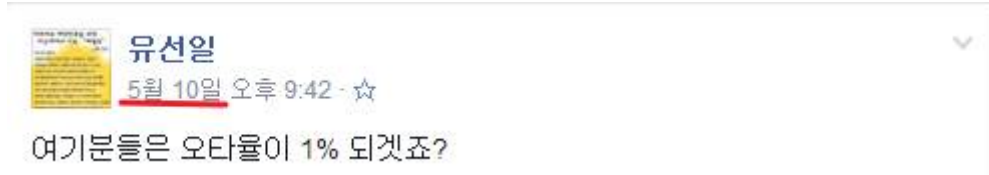


Picture(10) Create new tab with added URL

3.2 Techniques in Our Application

3.2.1. Adding bookmark buttons

The url we need to add is contained within red underline as seen below picture(11).



Picture(11) Where underlined contains url

We add bookmark buttons onto newsfeed, timeline, and, sidebar, by using the codes in the below pictures.

```
jQuery('._5pcp').each(function() {
    //Get sub-root node
    var element = jQuery(this);
    if ( element.attr("class") == "_5pcp" ) {
        //Get facebook link
        var link = element.find('_5pcq').attr('href');
        var add="https://www.facebook.com";
        //if the link is grouped.
        if(link == undefined || link.substring(0,1)=='#'){
            return;
        }
        if(link.substring(0,4)!="http"){
            link = add + link;
        }
        //Add a bookmark button and pass the link to the button
        element.append(" · ")
        element.append( jQuery('<a>').attr('href','#').bind('click', { p1: link }, bookmark.addBookMark).append("#").hover(
            function(){
                // hover
                $(this).text("#");
            },
            function(){
                // dishover
                $(this).text("#");
            }
        ));
        //Mark this node
        element.addClass("mark");
    }
});
```

Picture(12) Adding bookmark buttons onto newsfeed

In newsfeed, since the URL is located at under _5pcp class, we get the URL link from this class using jQuery as seen in the picture(12) above. To mark the bookmark buttons, we append star figure by using 'Append Method.' After you add a bookmark, the star figure is changed into a colored star figure.

```

jQuery('._1_n.fsm.fwn.fcg').each(function() {
    //Get sub-root node
    var element = jQuery(this);
    if ( element.attr("class") == "_1_n fsm fwn fcg" ) {

        //Get facebook link
        var link = element.find('.uiLinkSubtle').attr('href');

        var add="https://www.facebook.com";

        //if the link is grouped.
        if(link == undefined || link.substring(0,1)=='#'){
            return;
        }
        //console.log('extracted link : ' + link);
        if(link.substring(0,4)!="http"){
            link = add + link;
        }

        //Add a bookmark button and pass the link to the button
        element.append(" · ")
        element.append( jQuery('<a>').attr('href','#').bind('click', { p1: link }, bookmark.addBookMark).append("★").hover(
            function(){
                // hover
                $(this).text("★");
            },
            function(){
                // dishover
                $(this).text("☆");
            })
        );//end of append.

        //Mark this node
        element.addClass("mark");
    }
});

```

Picture(13) Adding bookmark buttons to Timeline

We are not only adding the bookmark buttons onto Timeline by finding the link from "_1_n fsm fwn fcg" class, but also, adding the url onto Timeline with this method.

```

jQuery('#fbPhotoSnowliftAudienceSelector').each(function() {
    var element = jQuery(this);
    if ( element.attr("class") == "mls" ) {
        var star = jQuery('<a>').attr('href','#').bind('click', { p1: document.URL }, bookmark.addBookMark).append("★").hover(
            function(){
                // hover
                $(this).text("★");
            },
            function(){
                // dishover
                $(this).text("☆");
            })
        )
        element.append(" · ").append(star).addClass("mark");
    }
});

```

Picture(14) Adding bookmark buttons to Sidebar

In the same way as picture(12) and pictures(13), we add the bookmark buttons onto sidebar.

3.2.2. Showing a dialog when we click a bookmark button

After you click a bookmark button, it calls 'addBookMark' function. In 'addBookMark' function, the 'showDialog' function is called as seen below pictures.

```
addBookMark: function(param) {
    console.log("addBookMark() start!");

    // check connection
    while ( port == null ) {
        console.log("trying to reconnect!");
        port = chrome.runtime.connect({name: "facebookmark"});
    }

    // request to get tag list and will set dialog
    port.postMessage({action: "requestTags"});

    // show dialog and will add bookmark
    bookmark.showDialog(param)
}
```

Picture(15) Calling 'showDialog' function when we click the bookmark button

```
jQuery('body').append(
    "<div id='dialog-form' title='Add A Book Mark'><p class='validateTips'>Fill forms to add bookmark.</p><form><fieldset style='border-color:#FFFFFF;'><label for='facebookmark_title'>Title</label><br><input type='text' id='facebookmark_title' class='text ui-widget-content ui-corner-all'><br><label for='facebookmark_folder'>Folder Name</label><br><input type='text' id='facebookmark_folder' value='Default' class='text ui-widget-content ui-corner-all'></fieldset></form></div>"
);
jQuery("#dialog-form").dialog({
    autoOpen: true,
    height: 270,
    width: 350,
    modal: true,
    buttons: {
        "OK": function() {
            // step 3
            //if title is null
            if($("#facebookmark_title").val()==""){
                var toast=$().toastmessage('showWarningToast',"Fill in the title form.");
                return false;
            }

            port.postMessage({action:"addBookMark",Title: $("#facebookmark_title").val(), Tag:$("#facebookmark_folder").val(),Url: param.data.p1});
            $("#facebookmark_title").val("");
            $( this ).dialog( "close" );
        },
        Cancel: function() {
            // cancel
            $("#facebookmark_title").val("");
            $( this ).dialog( "close" );
        }
    },
    close: function() {
        // close
        $("#facebookmark_title").val("");
        $( this ).dialog( "close" );
    }
});
```

Picture(16) 'showDialog' function

As you can see in the picture(16), we can create a dialog (it can be seen in the picture(8)) by following the codes seen in the picture(16).

```

chrome.runtime.onConnect.addListener(function(port) {
  console.assert(port.name == "facebookmark");
  port.onMessage.addListener(function(request) {
    var ROOT_FOLDER_NAME = "FaceBookMark";

    /*
     * In case of that request action is to add bookmark, execute below
     */
    if ( request.action == "addBookMark" ){
      chrome.bookmarks.getChildren("2",function(topChildren){
        var rootId = "";
        // check if root folder is exist
        var existRootFolder = false;

        topChildren.forEach(function(topChild){
          //console.log("finding root.." + result.title );
          if ( topChild.title == ROOT_FOLDER_NAME && existRootFolder == false ) {
            rootId = topChild.id;
            existRootFolder = true;
          }
        });
      });
    }
  });
});

```

Picture(17) Listener for request to add a bookmark

In dialog, the ok button plays the role to broadcast all the data which has titles, name of directories, and, urls to all component with 'Chrome message passing.' At that time, as you can see in the picture(17), the 'onMessage' listener that registered in background.js catches a broadcasted message. Then, it creates a bookmark with the data of message. Chrome has restricted to use most Chrome's API in content script, but we can solve this problem with this method.

See https://developer.chrome.com/extensions/content_scripts.

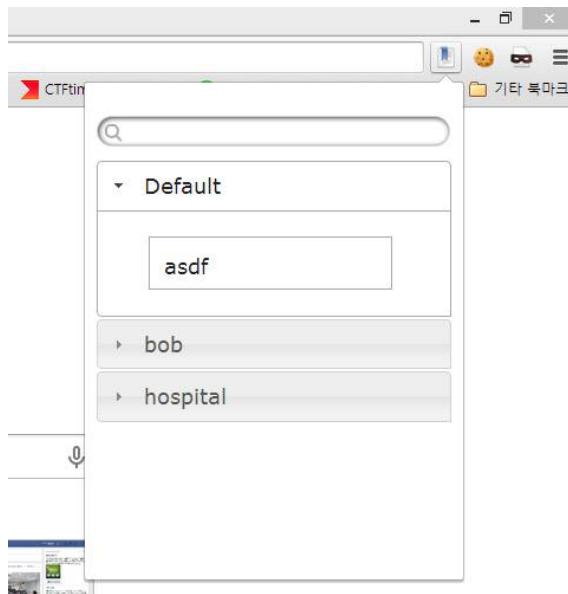
```

if ( existTag && tagId != "" ){
  // add there
  chrome.bookmarks.create({'parentId':tagId,'title':request.Title,'url':request.Url}, function(bookmark){
    //console.log("tag exist! added bookmark : " + bookmark.title);
    port.postMessage({action:"addBookMarkResponse"});
  });
} else {
  // tag does not exist
  // create folder and add there
  chrome.bookmarks.create({'parentId':rootId,'title':request.Tag}, function(tag){
    chrome.bookmarks.create({'parentId':tag.id,'title':request.Title,'url':request.Url}, function(bookmark){
      //console.log("tag not exist! added bookmark : " + bookmark.title);
      port.postMessage({action:"addBookMarkResponse"});
    });
  });
}
}

```

Picture(18) Add a bookmark into a folder

Our management method of bookmark is to make a directory, 'FaceBookMark,' and creates a bookmark with data of caught message. We introduce a concept, *tag*, as directory management. The concept is that all bookmark should have only one tag, having "Default" tag basically. The tag is the same as the directory name of message; because, tag is implemented just as sub-directory of 'FaceBookMark' directory. If the tag of caught message doesn't exist, we need to make a new directory that has the tag of the message and create a bookmark with data of caught message.



Picture(19) Browser action when you click application's icon

If you click application's icon, you can see the pop-up page as seen in the upper picture(19). This pop-up page is divided into 3 sections: Search-bar that can do real time search, Auto-complete, and, one accordion; and, this plays the role as displays collapsible content panels for presenting information in a limited amount of space, that shows search results.

```

/*
 * Listing
 * get a new section contents
 */
function getSectionContents(bookmarkList, query) {
    var section_root = jQuery('<ol>').addClass('section_content');
    // section contents layout
    for ( var i = 0; i < bookmarkList.length; i++ ){
        if ( query == "" || bookmarkList[i].title.indexOf(query) >= 0 ){
            section_root.append(
                jQuery('<li>').attr('class','ui-widget-content').append(
                    jQuery('<div>').text(bookmarkList[i].title).css('font-size','1em').add(
                        jQuery('<input>').attr('type','hidden').val(bookmarkList[i].url)
                    )
                )
            )
        }
    }
    }//end of if
} //end of for.
section_root.selectable({
    filter:'li',
    selected: function(event, ui) {
        chrome.tabs.create({url:${ui.selected}.find("input").val()})
    }
});
return section_root;
}

```

Picture(20) The core function which sets contents of accordion

If there is no input in search-bar in accordion, the accordion shows all the bookmarks; but, if there is any input exists, it shows all the bookmarks by classifying them with directory that includes matching string. If you click the accordion panel, it creates a new tab with the bookmark's url.

```
function setAccordion(query) {
    var ROOT_FOLDER_NAME = "FaceBookMark";
    chrome.bookmarks.getChildren("2",function(topChildren){
        var rootId = "";
        // check if root folder is exist
        var existRootFolder = false;

        topChildren.forEach(function(topChild){
            //console.log("finding root.." + result.title );
            if ( topChild.title == ROOT_FOLDER_NAME && existRootFolder == false ) {
                rootId = topChild.id;
                existRootFolder = true;
            }
        });

        // add sections if it is not null that returned bookmark list
        chrome.bookmarks.getSubTree(rootId, function(tagList) {
            // setAutocomplete
            setAutocomplete(tagList[0]);
            // traverse all tags.
            for ( var i = 0; i < tagList[0].children.length; i++ ){
```

Picture(21) Set materials for auto-complete

When we parse the whole bookmark to accordion, we saved the whole bookmark data to somewhere to implement auto-complete function in search-bar.



Picture(22) Demonstration for auto complete

We use jQuery UI including accordion and auto-complete, and use selectable function; and, this is also included in jQuery UI, in accordion panel.

Chapter 4 Future plan of developing Facebook Bookmark

Since Google Chrome only offers the bookmark manager that enables us only edit and delete through it, now we're working on a project for more functions for the 'Facebook Bookmark'; for instance, we can apply bookmark function by using keyboard shortcut, and, by typing enter key in dialog, we can add a bookmark and edit/delete in application.

Chapter 5 References

<https://github.com/akquinet/jquery-toastmessage-plugin/wiki>

<https://developer.chrome.com/extensions>