原本澎澎的code>>>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Photo Editor</title>

<script>

// 影像編輯功能

var ctx;

window.onload=function(){

    ctx=document.getElementById("cvs").getContext("2d"); // 取得 canvas 相對應的 context 物件

    // 載入圖片

    loadImage("landscape.jpg");

};

function loadImage(src){

    var img=new Image(); // <img/>

    img.src=src;

    img.onload=function(){

        // 確定圖檔載入完成之後，才畫進 canvas 裡面

        ctx.drawImage(this, 0, 0, ctx.canvas.width, ctx.canvas.height);

    };

}

function invert(){

    // 取得像素資料

    var pixels=ctx.getImageData(0, 0, 600, 450);

    // 以像素為單位處理資料

    for(var i=0;i<pixels.data.length;i+=4){

        pixels.data[i]=255-pixels.data[i]; // r

        pixels.data[i+1]=255-pixels.data[i+1]; // g

        pixels.data[i+2]=255-pixels.data[i+2]; // b

    }

    // 把處理完的像素資料放回 canvas 裡面

    ctx.putImageData(pixels, 0, 0);

}

</script>

</head>

<body>

    <div>

        <button onclick="invert();">Invert</button>

    </div>

    <canvas id="cvs" width="600" height="450"

        style="border:1px solid black"></canvas>

</body>

</html>

2.

// 以下是Google Drive 相關的程式

function(){

}

但是使用者根本沒依據啊!!

沒授權

所以將前一範例的:

<script async defer src="https://apis.google.com/js/api.js"

onload="this.onload=function(){};handleClientLoad()"

onreadystatechange="if (this.readyState === 'complete') this.onload()">

</script>

放進來

可改成喜歡的名字!!!

// 以下是Google Drive 相關的程式

function initGoogleAPI(){

}

function openFile(){

// 使用者授權了沒??? // 去拿前一個範例的code

}

</script>

<script async defer src="https://apis.google.com/js/api.js"

onload="this.onload=function(){};initGoogleAPI()"

onreadystatechange="if (this.readyState === 'complete') this.onload()">

</script>

3.

可多要依些授權範圍:

可以有很多SCOPE可以用:

<https://developers.google.com/drive/v3/web/about-auth>

|  |  |
| --- | --- |
| https://www.googleapis.com/auth/drive | Full, permissive scope to access all of a user's files, excluding the [Application Data folder](https://developers.google.com/drive/web/appdata). Request this scope only when it is strictly necessary. |

這個權限最高(所有這個權限)>> 建議開發時全開~ 但是之後主攻能做完再慢慢限縮!!! 最後給使用者最低權限

這樣的開發流程是好的，減少干擾!

4. 問題:

未授權也應該要可以用(從本機端打開檔)，如果不需要不用做!

不要忘了

已授權的APP可以到google account授權取消

目前:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Photo Editor</title>

<script>

// 影像編輯功能

var ctx;

window.onload=function(){

    ctx=document.getElementById("cvs").getContext("2d"); // 取得 canvas 相對應的 context 物件

    // 載入圖片

    loadImage("landscape.jpg");

};

function loadImage(src){

    var img=new Image(); // <img/>

    img.src=src;

    img.onload=function(){

        // 確定圖檔載入完成之後，才畫進 canvas 裡面

        ctx.drawImage(this, 0, 0, ctx.canvas.width, ctx.canvas.height);

    };

}

function invert(){

    // 取得像素資料

    var pixels=ctx.getImageData(0, 0, 600, 450);

    // 以像素為單位處理資料

    for(var i=0;i<pixels.data.length;i+=4){

        pixels.data[i]=255-pixels.data[i]; // r

        pixels.data[i+1]=255-pixels.data[i+1]; // g

        pixels.data[i+2]=255-pixels.data[i+2]; // b

    }

    // 把處理完的像素資料放回 canvas 裡面

    ctx.putImageData(pixels, 0, 0);

}

// 以下是Google Drive 相關的程式

// step3

function initGoogleAPI(){

gapi.load('client:auth2', initClient);

}

// step 7 (可改成const 用大寫是一個習慣)

const CLIENT\_ID = '674608619453-e8igkl5mpvkuei35g0odekiqa73cgl36.apps.googleusercontent.com';

const API\_KEY = 'AIzaSyDHI95QsrnD1ynZcBaql876xOpSy\_eoBBM';

// step 8 原本的scope的權限很爛，只有基本metadata且是read only，所以我們要多要依些權限!!!看doc: https://developers.google.com/drive/v3/web/about-auth

const SCOPES = 'https://www.googleapis.com/auth/drive'; //這裡因開發先全開，之後再慢慢關!

// step 10

// 為了要儲存授權狀態!另個全預變數方便未來使用 (使用者不一定馬上需要用啊，因為可能想開本機端檔)

let auth = null;

// step4 (Callback概念!)

function initClient(){

// step5 先放google的再改

// 偵測授權狀態!

gapi.client.init({

// step6 補上這些全域變數，不用寫在最上面，只要在最外層即可!

apiKey: API\_KEY,

clientId: CLIENT\_ID,

// discoveryDocs: DISCOVERY\_DOCS, // 其實這裡刪掉也沒差!

scope: SCOPES // 授權範圍! (看上面!用網址當作授權的名字)

}).then(function () {

// >>>> .then這裡已得到偵測的結果

// Listen for sign-in state changes.

// 未來若有任何授權狀態的變化，呼叫updateSigninStatus!

// step 11: 把Assign給auth

auth = gapi.auth2.getAuthInstance();

// Handle the initial sign-in state.

// 授權狀態的資料:gapi.auth2.getAuthInstance().isSignedIn.get() >> 會回傳是否已授權(True or False)(背後都會去連到google server!)

// step 9

// 以下因為不一定要馬上授權，所以先刪掉! 但是授權狀態要保持，需要時還是要知道!

// // 根據授權狀態做不同的處裡!

// updateSigninStatus(gapi.auth2.getAuthInstance().isSignedIn.get()); // 一有變更就來update!!!

// authorizeButton.onclick = handleAuthClick; //基本的javascript onclick

// signoutButton.onclick = handleSignoutClick;

});

}

// step1

function openFile(){

// 使用者授權了沒??? // 去拿前一個範例的code

// step 12:

if(auth.isSignedIn.get()){ // 已授權

// 打開雲端硬碟的檔案選擇器

alert("ok!!");

}else{ //未授權

auth.signIn().then(function(){

// 使用者確定授權後的工作!

alert("OK!");

});

}

}

</script>

<!-- step2 -->

<script async defer src="https://apis.google.com/js/api.js"

onload="this.onload=function(){};initGoogleAPI()"

onreadystatechange="if (this.readyState === 'complete') this.onload()">

</script>

</head>

<body>

<div>

<button onclick="openFile();">Open File</button>

        <button onclick="invert();">Invert</button>

    </div>

    <canvas id="cvs" width="600" height="450"

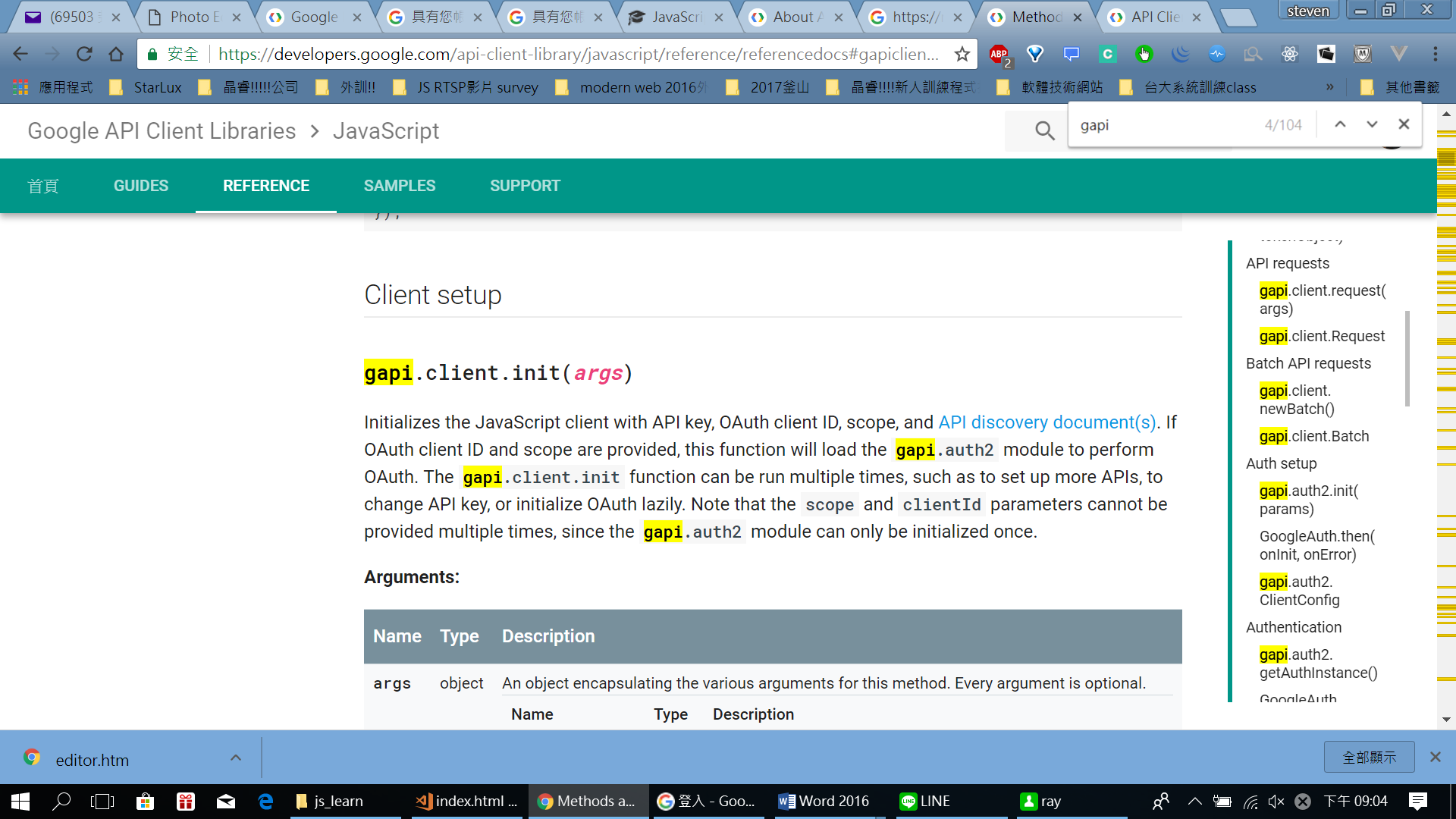
        style="border:1px solid black"></canvas>

</body>

</html>

<https://developers.google.com/api-client-library/>

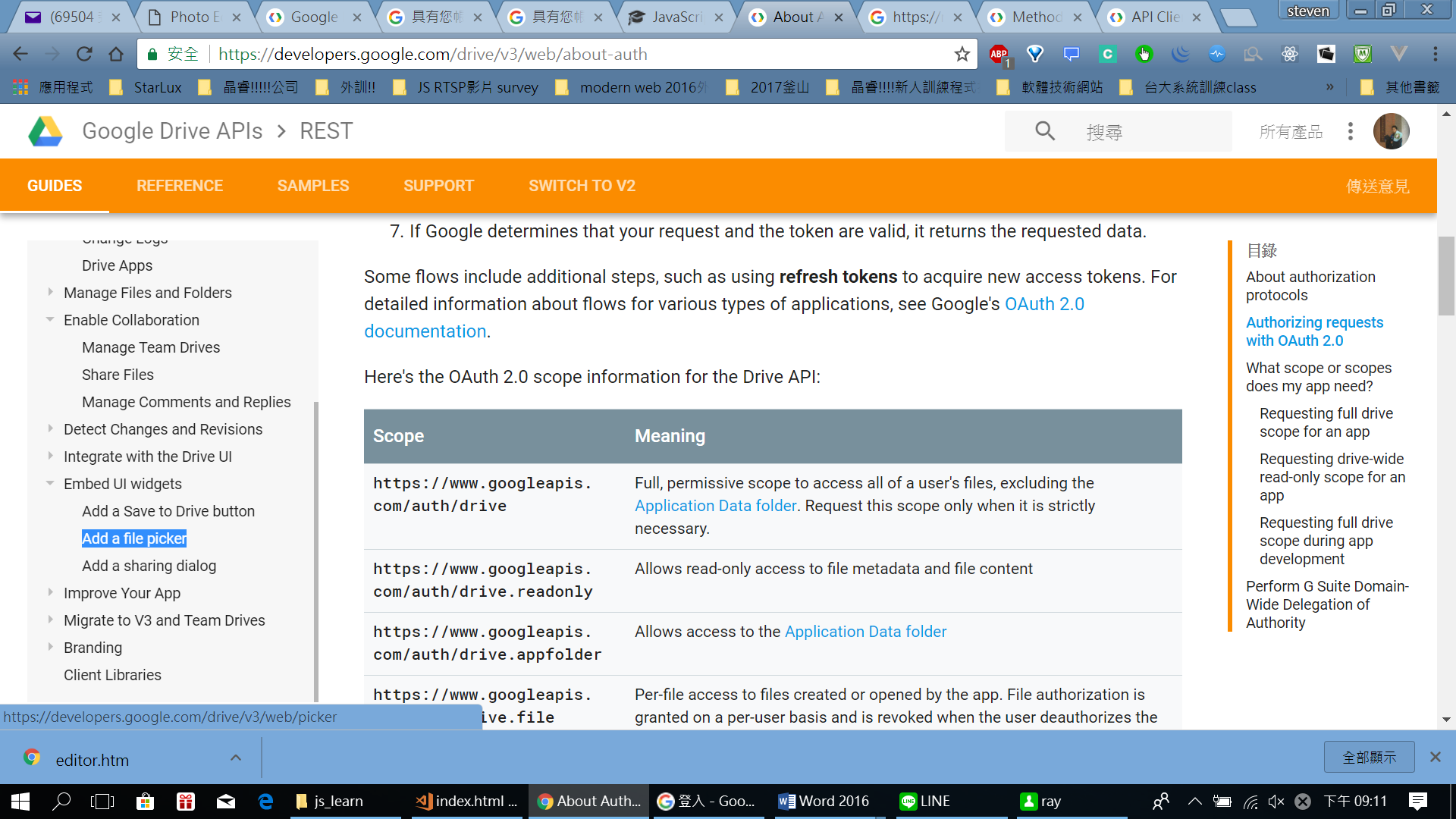
<https://developers.google.com/api-client-library/javascript/reference/referencedocs>

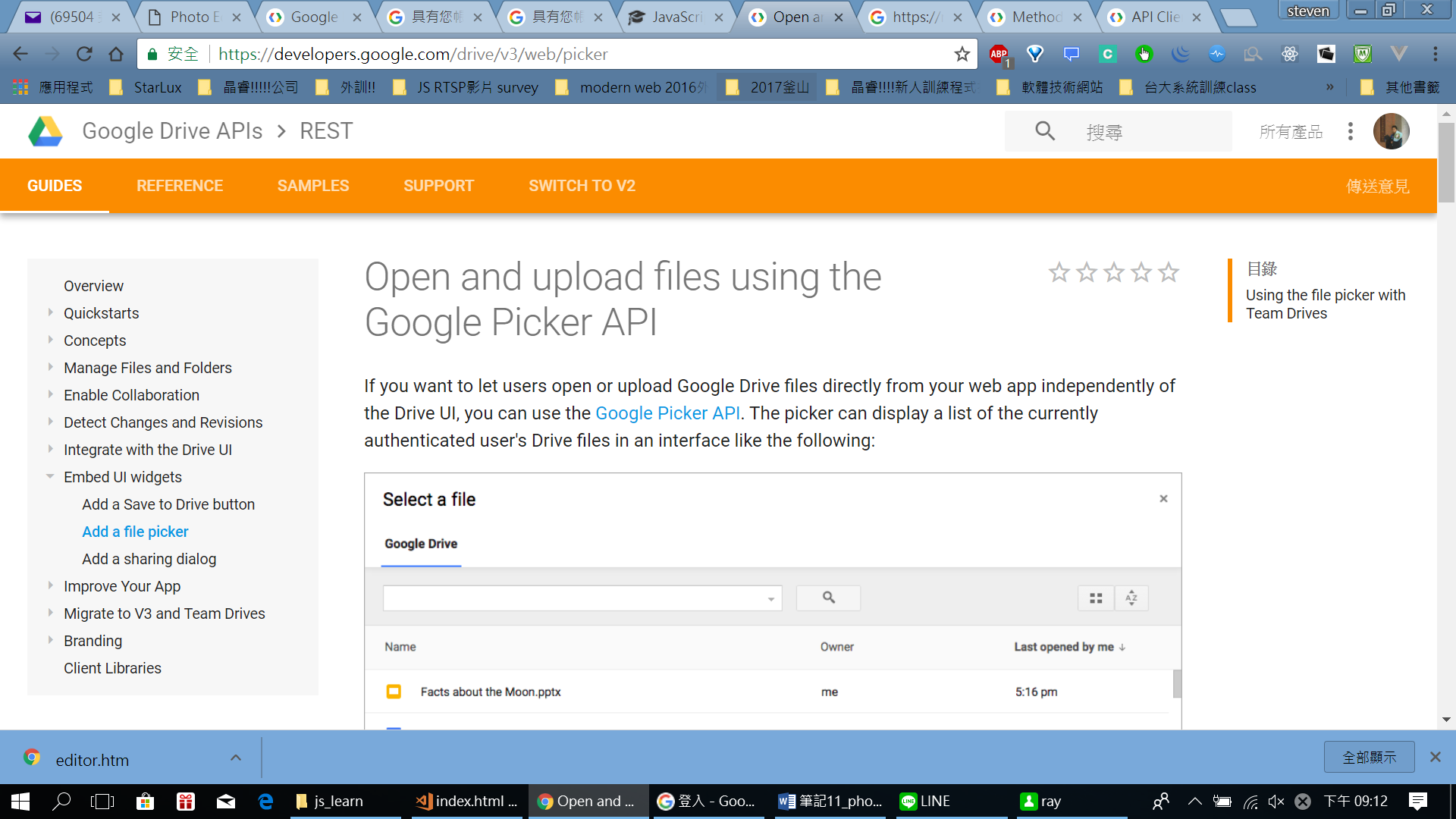


>>>>>>>>

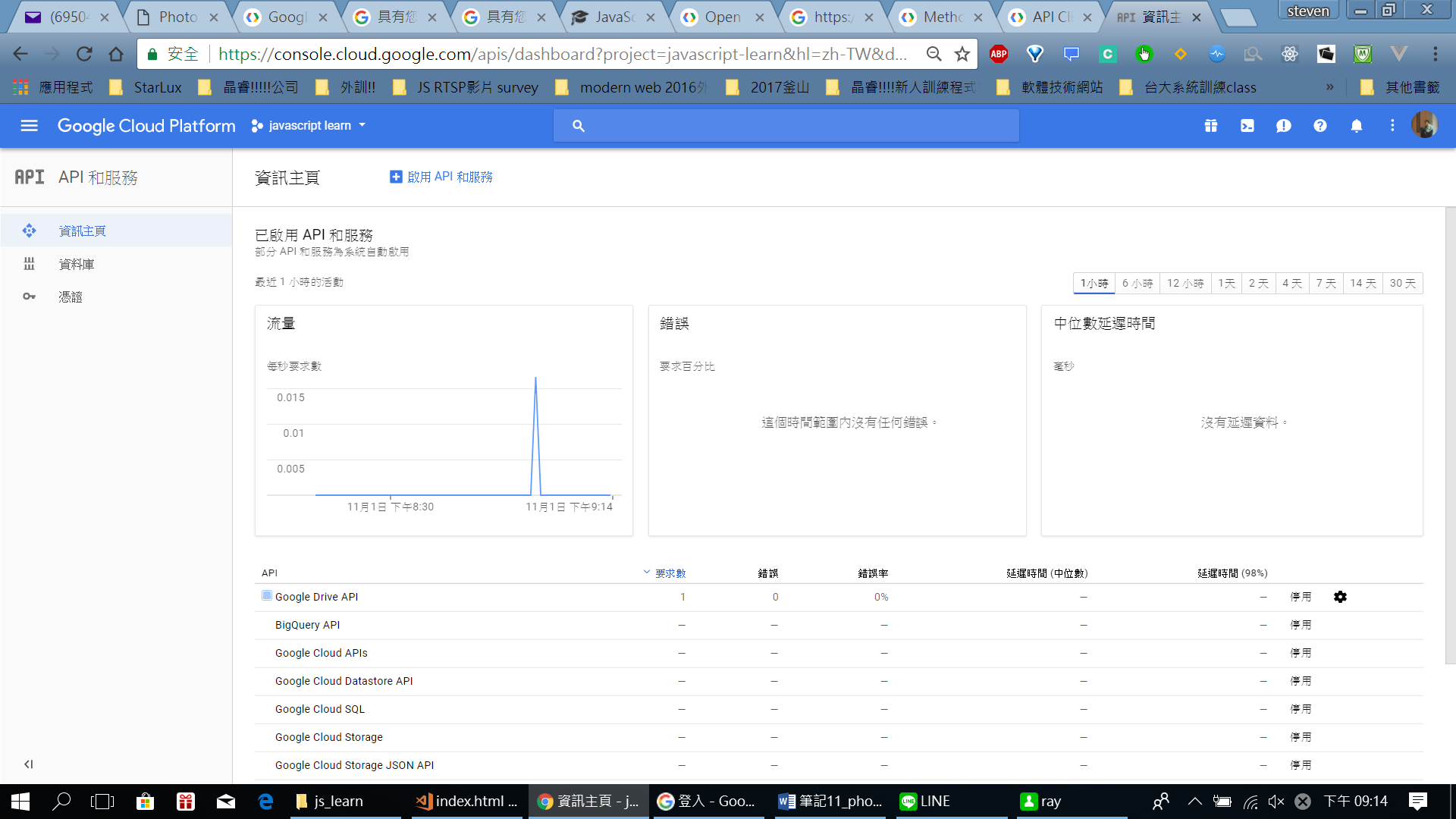
接下來因為如何開放介面>做driver的檔案選擇>> file picker

<還可以串youtube等等>

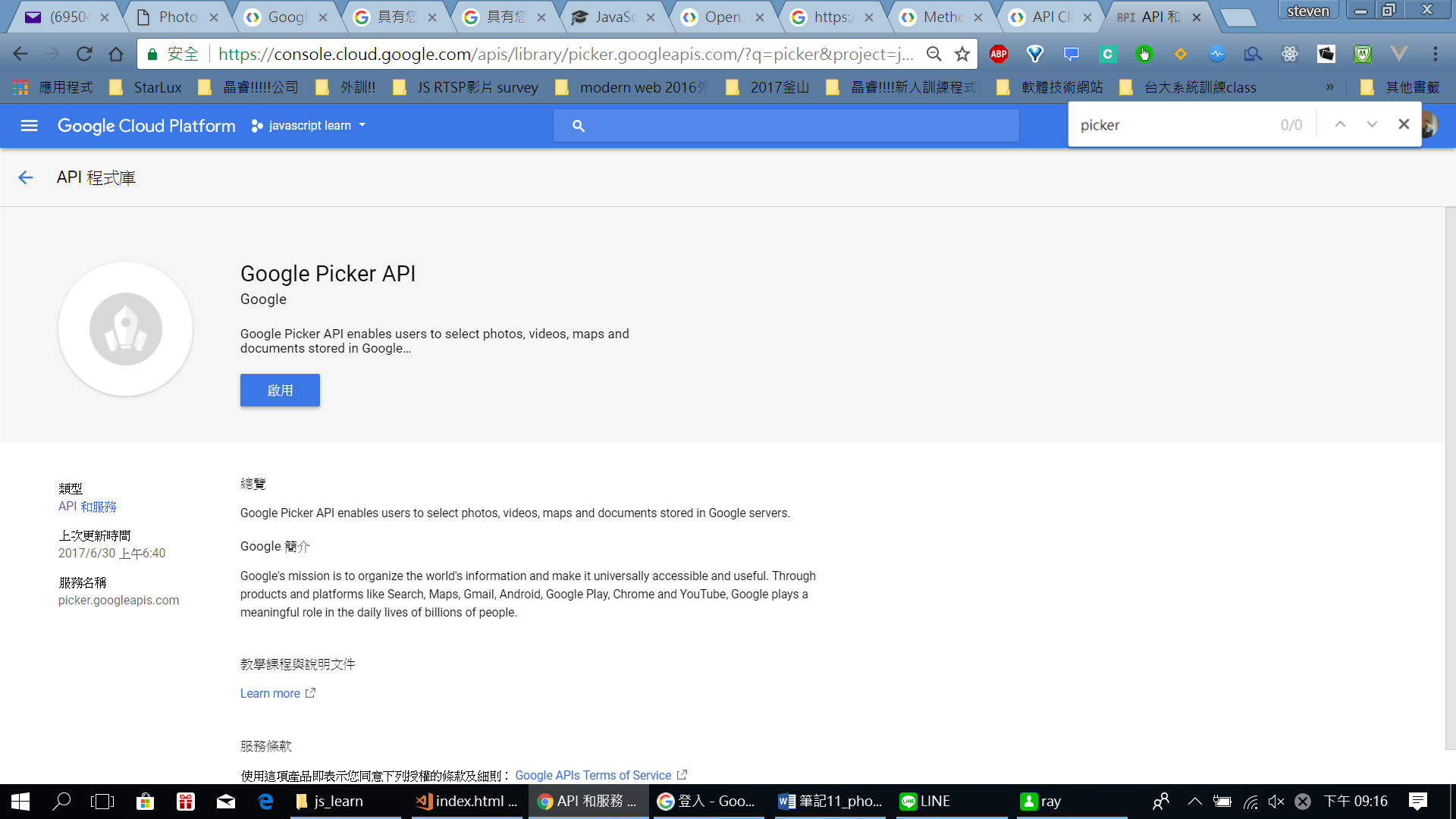




先到:



啟用picker API



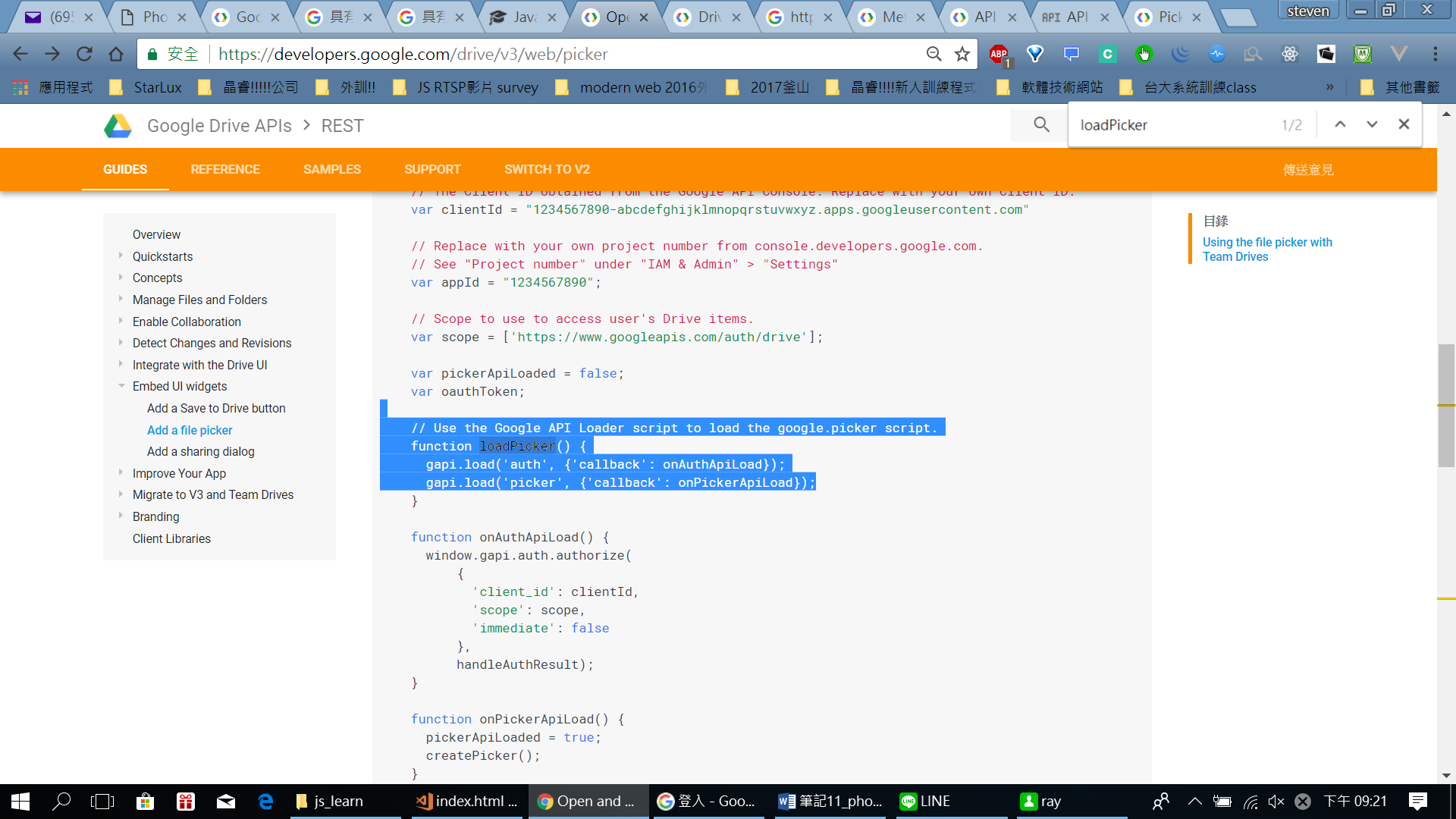
回到API

Sample:

<https://developers.google.com/picker/docs/>

<https://developers.google.com/drive/v3/web/picker> (功能較多，用這個好了!)

所以可以開始改:



目前Code:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Photo Editor</title>

<script>

// 影像編輯功能

var ctx;

window.onload=function(){

    ctx=document.getElementById("cvs").getContext("2d"); // 取得 canvas 相對應的 context 物件

    // 載入圖片

    loadImage("landscape.jpg");

};

function loadImage(src){

    var img=new Image(); // <img/>

    img.src=src;

    img.onload=function(){

        // 確定圖檔載入完成之後，才畫進 canvas 裡面

        ctx.drawImage(this, 0, 0, ctx.canvas.width, ctx.canvas.height);

    };

}

function invert(){

    // 取得像素資料

    var pixels=ctx.getImageData(0, 0, 600, 450);

    // 以像素為單位處理資料

    for(var i=0;i<pixels.data.length;i+=4){

        pixels.data[i]=255-pixels.data[i]; // r

        pixels.data[i+1]=255-pixels.data[i+1]; // g

        pixels.data[i+2]=255-pixels.data[i+2]; // b

    }

    // 把處理完的像素資料放回 canvas 裡面

    ctx.putImageData(pixels, 0, 0);

}

// 以下是Google Drive 相關的程式

// step3

function initGoogleAPI(){

// step 13:

gapi.load('client:auth2', initClient);

gapi.load('picker');

}

// step 7 (可改成const 用大寫是一個習慣)

const CLIENT\_ID = '674608619453-e8igkl5mpvkuei35g0odekiqa73cgl36.apps.googleusercontent.com';

const API\_KEY = 'AIzaSyDHI95QsrnD1ynZcBaql876xOpSy\_eoBBM';

// step 8 原本的scope的權限很爛，只有基本metadata且是read only，所以我們要多要依些權限!!!看doc: https://developers.google.com/drive/v3/web/about-auth

const SCOPES = 'https://www.googleapis.com/auth/drive'; //這裡因開發先全開，之後再慢慢關!

// step 10

// 為了要儲存授權狀態!另個全預變數方便未來使用 (使用者不一定馬上需要用啊，因為可能想開本機端檔)

let auth = null;

// step4 (Callback概念!)

function initClient(){

// step5 先放google的再改

// 偵測授權狀態!

gapi.client.init({

// step6 補上這些全域變數，不用寫在最上面，只要在最外層即可!

apiKey: API\_KEY,

clientId: CLIENT\_ID,

// discoveryDocs: DISCOVERY\_DOCS, // 其實這裡刪掉也沒差!

scope: SCOPES // 授權範圍! (看上面!用網址當作授權的名字)

}).then(function () {

// >>>> .then這裡已得到偵測的結果

// Listen for sign-in state changes.

// 未來若有任何授權狀態的變化，呼叫updateSigninStatus!

// step 11: 把Assign給auth 只取到()

auth = gapi.auth2.getAuthInstance();

// Handle the initial sign-in state.

// 授權狀態的資料:gapi.auth2.getAuthInstance().isSignedIn.get() >> 會回傳是否已授權(True or False)(背後都會去連到google server!)

// step 9

// 以下因為不一定要馬上授權，所以先刪掉! 但是授權狀態要保持，需要時還是要知道!

// // 根據授權狀態做不同的處裡!

// updateSigninStatus(gapi.auth2.getAuthInstance().isSignedIn.get()); // 一有變更就來update!!!

// authorizeButton.onclick = handleAuthClick; //基本的javascript onclick

// signoutButton.onclick = handleSignoutClick;

});

}

// step1

function openFile(){

// 使用者授權了沒??? // 去拿前一個範例的code

// step 12:

if(auth.isSignedIn.get()){ // 已授權

// 打開雲端硬碟的檔案選擇器

// alert("ok!!");

createPicker();

}else{ //未授權

auth.signIn().then(function(){

// 使用者確定授權後的工作!

alert("OK!");

});

}

}

// Create and render a Picker object for searching images.

function createPicker() {

// if (oauthToken) { // 確保使用者已授權但是因為在上面的if else已經確定授權才進來了!

var view = new google.picker.View(google.picker.ViewId.DOCS); //秀出google drive文件

view.setMimeTypes("image/png,image/jpeg,image/jpg");

var picker = new google.picker.PickerBuilder() // 很關鍵，設定檔案選擇器的基本功能

// .enableFeature(google.picker.Feature.NAV\_HIDDEN) // 開啟功能:瀏覽輛藏起來不要好了..

// .enableFeature(google.picker.Feature.MULTISELECT\_ENABLED) // 可以多選，這裡影像先單選就好不要

// .setAppId(appId) // appId 就是專案ID (上方CLIENT-ID前面的專案ID 674608619453)

.setAppId("674608619453") // appId 就是專案ID (上方CLIENT-ID前面的專案ID 674608619453)

.setOAuthToken(auth.currentUser.get().getAuthResponse().access\_token) // oauthToken: 授權完發現你是一個有被授權的人，為了讓之後能認得你是誰發了token給你(暫時的token)

.addView(view) // 建立上面那個view

.addView(new google.picker.DocsUploadView()) // 建立上傳的view (幫我們寫好就不用煩惱啦!)

.setDeveloperKey(API\_KEY)

.setCallback(pickerCallback) //這個一定要寫 開一個介面使用者選了檔案完要從哪裡接下去!

.build();

picker.setVisible(true);

// }

}

function pickerCallback(){

}

</script>

<!-- step2 -->

<script async defer src="https://apis.google.com/js/api.js"

onload="this.onload=function(){};initGoogleAPI()"

onreadystatechange="if (this.readyState === 'complete') this.onload()">

</script>

</head>

<body>

<div>

<button onclick="openFile();">Open File</button>

        <button onclick="invert();">Invert</button>

    </div>

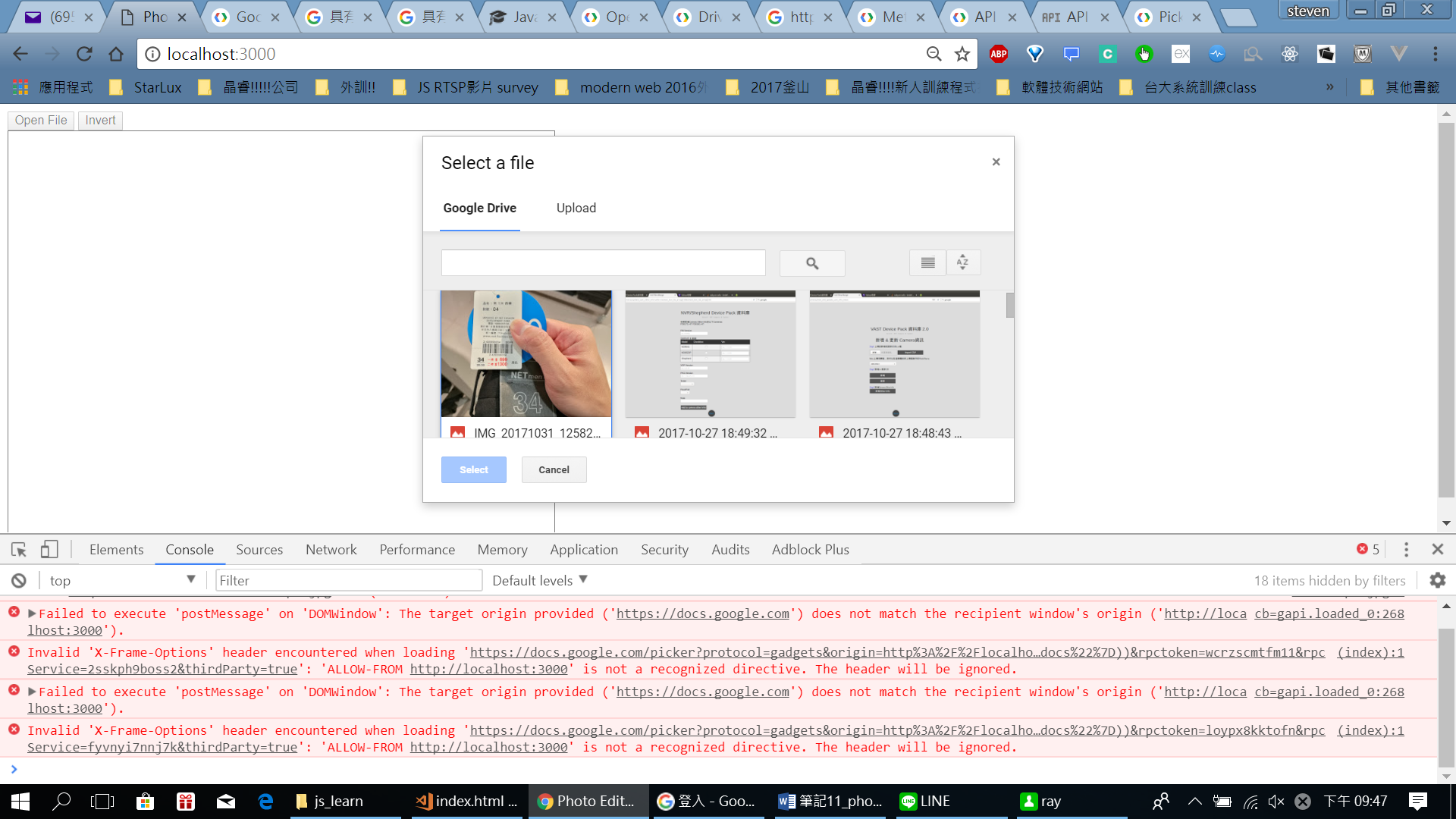
    <canvas id="cvs" width="600" height="450"

        style="border:1px solid black"></canvas>

</body>

</html>

就成功拉!!:



// 使用者選了檔案後 做的是無論是選了檔還是關閉! 使用者按XX 依樣會callback!

// A simple callback implementation.

function pickerCallback(data) {

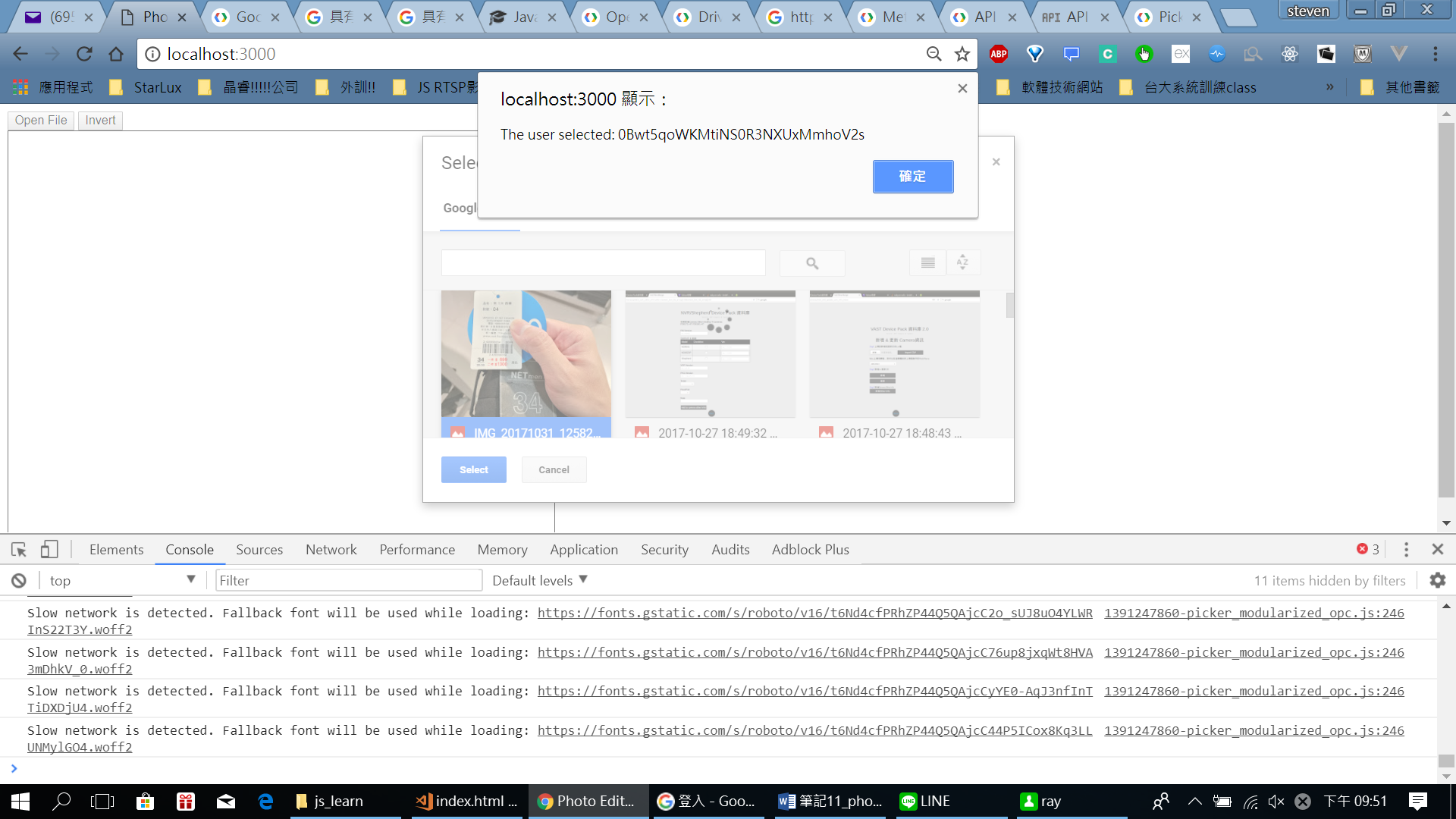
if (data.action == google.picker.Action.PICKED) {

var fileId = data.docs[0].id; // 選了檔案把 id抓出來

alert('The user selected: ' + fileId);

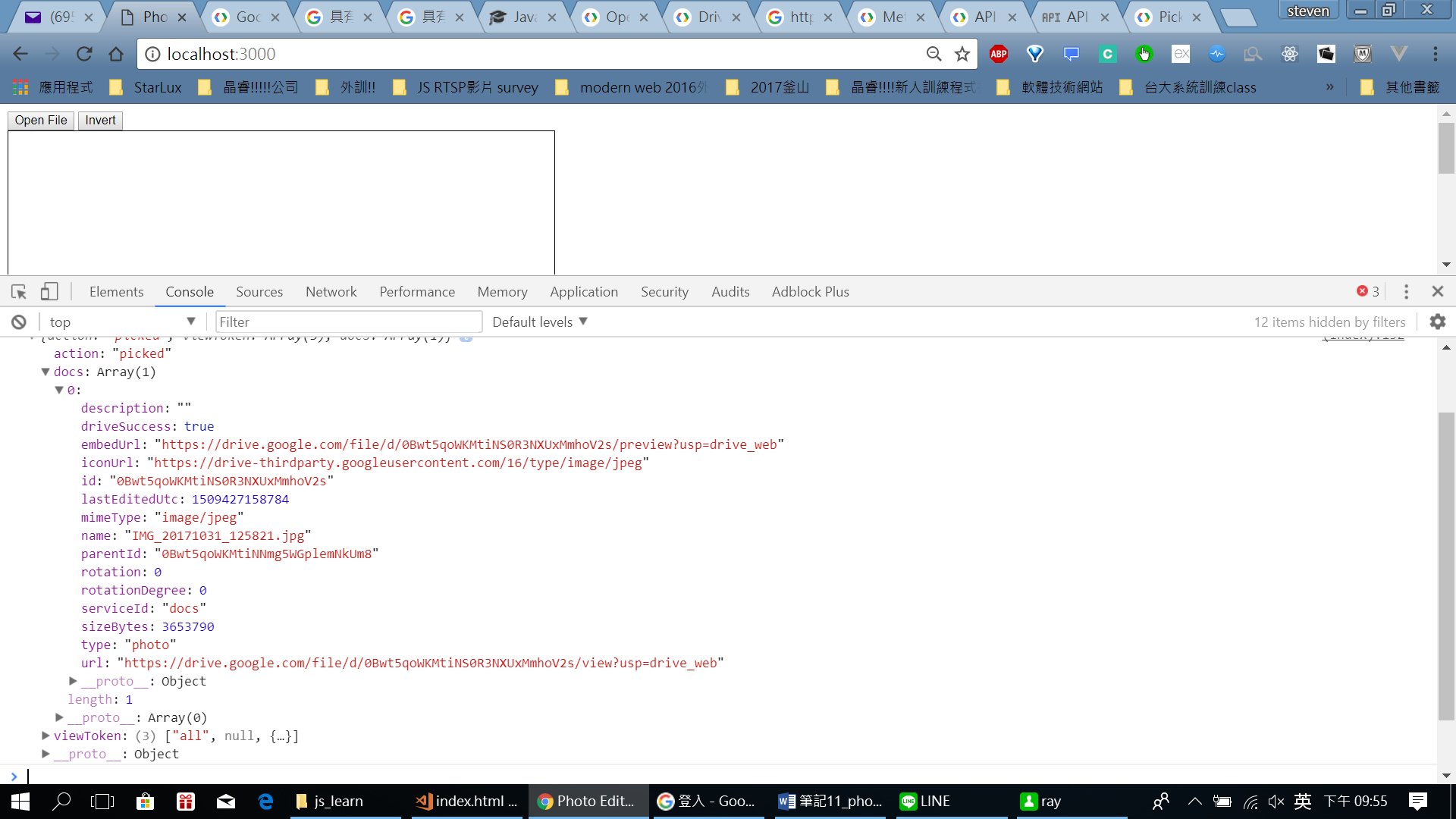
}

}



可印出data

:



目前:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Photo Editor</title>

<script>

// 影像編輯功能

var ctx;

window.onload=function(){

    ctx=document.getElementById("cvs").getContext("2d"); // 取得 canvas 相對應的 context 物件

    // 載入圖片

    loadImage("landscape.jpg");

};

function loadImage(src){

    var img=new Image(); // <img/>

    img.src=src;

    img.onload=function(){

        // 確定圖檔載入完成之後，才畫進 canvas 裡面

        ctx.drawImage(this, 0, 0, ctx.canvas.width, ctx.canvas.height);

    };

}

function invert(){

    // 取得像素資料

    var pixels=ctx.getImageData(0, 0, 600, 450);

    // 以像素為單位處理資料

    for(var i=0;i<pixels.data.length;i+=4){

        pixels.data[i]=255-pixels.data[i]; // r

        pixels.data[i+1]=255-pixels.data[i+1]; // g

        pixels.data[i+2]=255-pixels.data[i+2]; // b

    }

    // 把處理完的像素資料放回 canvas 裡面

    ctx.putImageData(pixels, 0, 0);

}

// 以下是Google Drive 相關的程式

// step3

function initGoogleAPI(){

// step 13:

gapi.load('client:auth2', initClient);

gapi.load('picker');

}

// step 7 (可改成const 用大寫是一個習慣)

const CLIENT\_ID = '674608619453-e8igkl5mpvkuei35g0odekiqa73cgl36.apps.googleusercontent.com';

const API\_KEY = 'AIzaSyDHI95QsrnD1ynZcBaql876xOpSy\_eoBBM';

// step 8 原本的scope的權限很爛，只有基本metadata且是read only，所以我們要多要依些權限!!!看doc: https://developers.google.com/drive/v3/web/about-auth

const SCOPES = 'https://www.googleapis.com/auth/drive'; //這裡因開發先全開，之後再慢慢關!

// step 10

// 為了要儲存授權狀態!另個全預變數方便未來使用 (使用者不一定馬上需要用啊，因為可能想開本機端檔)

let auth = null;

// step4 (Callback概念!)

function initClient(){

// step5 先放google的再改

// 偵測授權狀態!

gapi.client.init({

// step6 補上這些全域變數，不用寫在最上面，只要在最外層即可!

apiKey: API\_KEY,

clientId: CLIENT\_ID,

// discoveryDocs: DISCOVERY\_DOCS, // 其實這裡刪掉也沒差!

scope: SCOPES // 授權範圍! (看上面!用網址當作授權的名字)

}).then(function () {

// >>>> .then這裡已得到偵測的結果

// Listen for sign-in state changes.

// 未來若有任何授權狀態的變化，呼叫updateSigninStatus!

// step 11: 把Assign給auth 只取到()

auth = gapi.auth2.getAuthInstance();

// Handle the initial sign-in state.

// 授權狀態的資料:gapi.auth2.getAuthInstance().isSignedIn.get() >> 會回傳是否已授權(True or False)(背後都會去連到google server!)

// step 9

// 以下因為不一定要馬上授權，所以先刪掉! 但是授權狀態要保持，需要時還是要知道!

// // 根據授權狀態做不同的處裡!

// updateSigninStatus(gapi.auth2.getAuthInstance().isSignedIn.get()); // 一有變更就來update!!!

// authorizeButton.onclick = handleAuthClick; //基本的javascript onclick

// signoutButton.onclick = handleSignoutClick;

});

}

// step1

function openFile(){

// 使用者授權了沒??? // 去拿前一個範例的code

// step 12:

if(auth.isSignedIn.get()){ // 已授權

// 打開雲端硬碟的檔案選擇器

// alert("ok!!");

createPicker();

}else{ //未授權

auth.signIn().then(function(){

// 使用者確定授權後的工作!

alert("OK!");

});

}

}

// Create and render a Picker object for searching images.

function createPicker() {

// if (oauthToken) { // 確保使用者已授權但是因為在上面的if else已經確定授權才進來了!

var view = new google.picker.View(google.picker.ViewId.DOCS); //秀出google drive文件

view.setMimeTypes("image/png,image/jpeg,image/jpg");

var picker = new google.picker.PickerBuilder() // 很關鍵，設定檔案選擇器的基本功能

// .enableFeature(google.picker.Feature.NAV\_HIDDEN) // 開啟功能:瀏覽輛藏起來不要好了..

// .enableFeature(google.picker.Feature.MULTISELECT\_ENABLED) // 可以多選，這裡影像先單選就好不要

// .setAppId(appId) // appId 就是專案ID (上方CLIENT-ID前面的專案ID 674608619453)

.setAppId("674608619453") // appId 就是專案ID (上方CLIENT-ID前面的專案ID 674608619453)

.setOAuthToken(auth.currentUser.get().getAuthResponse().access\_token) // oauthToken: 授權完發現你是一個有被授權的人，為了讓之後能認得你是誰發了token給你(暫時的token)

.addView(view) // 建立上面那個view

.addView(new google.picker.DocsUploadView()) // 建立上傳的view (幫我們寫好就不用煩惱啦!)

.setDeveloperKey(API\_KEY)

.setCallback(pickerCallback) //這個一定要寫 開一個介面使用者選了檔案完要從哪裡接下去!

.build();

picker.setVisible(true);

// }

}

// 使用者選了檔案後 做的是無論是選了檔還是關閉! 使用者按XX 依樣會callback!

// A simple callback implementation.

function pickerCallback(data) {

// console.log(data)

if (data.action == google.picker.Action.PICKED) {

var fileId = data.docs[0].id; // 選了檔案把 id抓出來

alert('The user selected: ' + fileId);

downloadFile(fileId); // 把圖檔本身抓下來

}

}

function downloadFile(fileId){

// 稍微複雜.......... 下次再來!!!

}

</script>

<!-- step2 -->

<script async defer src="https://apis.google.com/js/api.js"

onload="this.onload=function(){};initGoogleAPI()"

onreadystatechange="if (this.readyState === 'complete') this.onload()">

</script>

</head>

<body>

<div>

<button onclick="openFile();">Open File</button>

        <button onclick="invert();">Invert</button>

    </div>

    <canvas id="cvs" width="600" height="450"

        style="border:1px solid black"></canvas>

</body>

</html>

>>>>>>>>>下次繼續:

<https://developers.google.com/picker/docs/reference>

可以看

DocsUploadView

等等View

項可以放MapsView 就會有地圖出來

Youtube等等 (但還需要Youtube授權!)