MAKALAH

PENDAFTARAN GOOGLE API

Oleh:

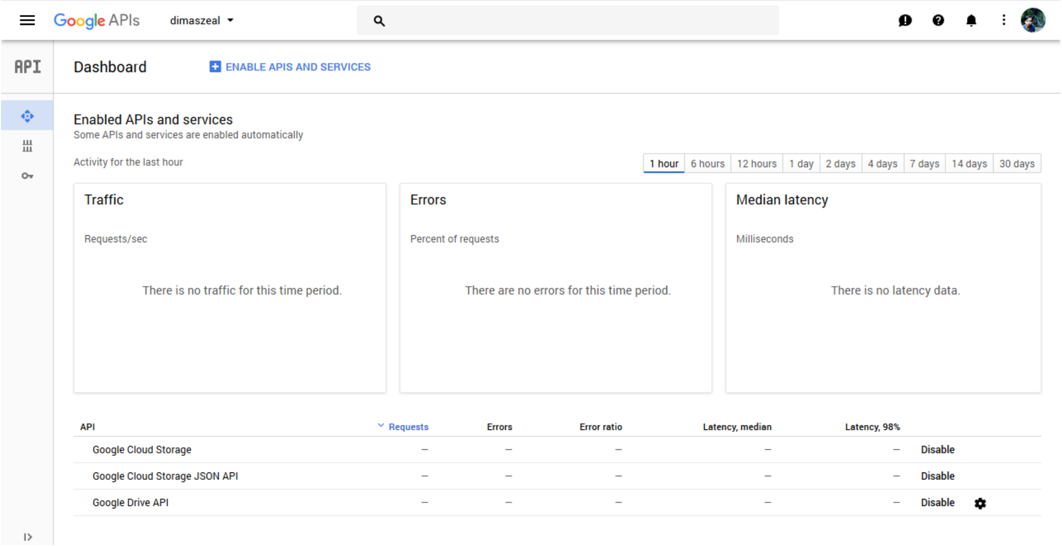
Dimas Luthfi Pratama Putra

14043020

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BHAYANGKARA SURABAYA
2017

Pendaftaran Google API

Untuk membuat sebuah project atau menggunakan Google API kita terlebih dahulu harus melakukan pendaftaran akun di google Developers. Untuk create akun google developer kita bisa mengakses halaman https://console.developers.google.com/project , maka akan muncul sebuah windows yang akan terlihat pada gambar dibawah ini :



Enabled APIs and services

Some APIs and services are enabled automatically

Activity for the last hour

Traffic

Requests/sec

There is no traffic for this time period.

Errors

Percent of requests

There are no errors for this time period.

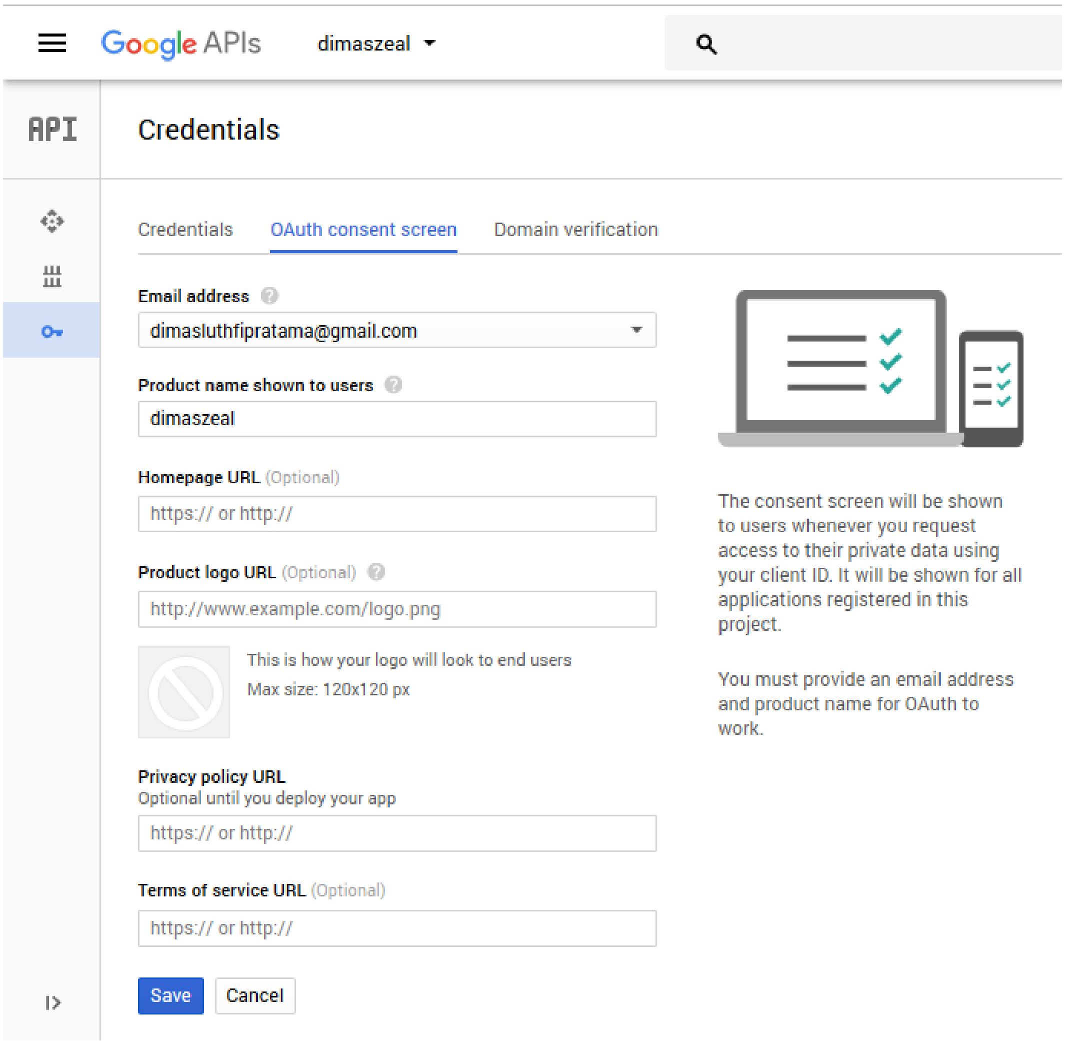
Median latency

Milliseconds

There is no latency data.

API	Requests	Errors	Error ratio	Latency, median	Latency, 98%	
Google Cloud Storage	–	–	–	–	–	Disable
Google Cloud Storage JSON API	–	–	–	–	–	Disable
Google Drive API	–	–	–	–	–	Disable

Pada langkah selanjutnya kita pilih layer persetujuan OAuth, maka akan muncul windows seperti gambar di bawah ini :



Email address
dimasluthfipratama@gmail.com

Product name shown to users
dimaszeal

Homepage URL (Optional)
https:// or http://

Product logo URL (Optional)
http://www.example.com/logo.png

This is how your logo will look to end users
Max size: 120x120 px

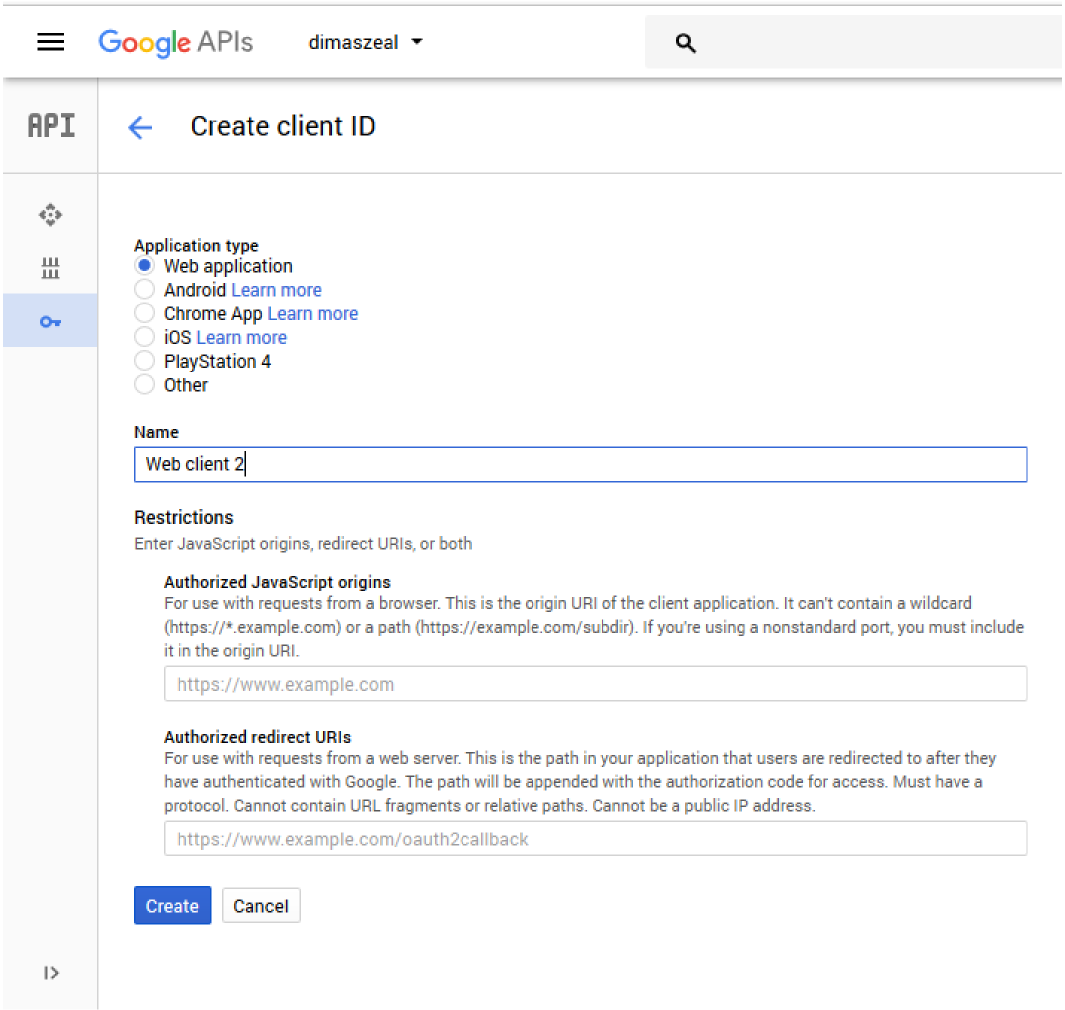
Privacy policy URL
Optional until you deploy your app
https:// or http://

Terms of service URL (Optional)
https:// or http://

The consent screen will be shown to users whenever you request access to their private data using your client ID. It will be shown for all applications registered in this project.

You must provide an email address and product name for OAuth to work.

Jika sudah muncul windows seperti gambar diatas, selanjutnya silahkan isi nama produk yang akan di tampilkan kepda pengguna lalu klik simpan. Jika berhasil maka akan muncul windows permintaan membuat id client seperti gambar di bawah ini :



Create client ID

Application type

● Web application
○ Android Learn more
○ Chrome App Learn more
○ iOS Learn more
○ PlayStation 4
○ Other

Name

Web client 2

Restrictions

Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

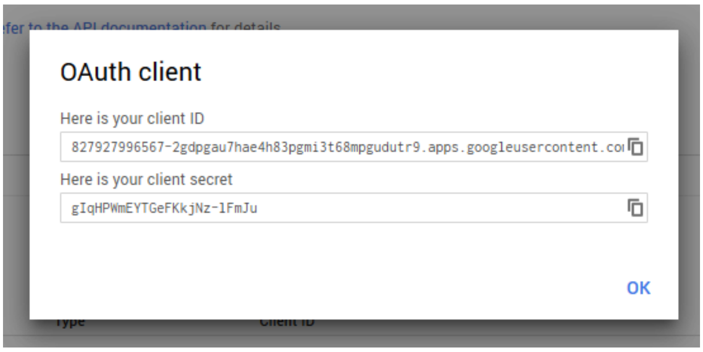
https://www.example.com

Authorized redirect URIs

For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

https://www.example.com/oauth2callback

Pilih jenis aplikasi yang akan digunakan, selanjutnya isi field nama jika sudah klik buat/create. Jika berhasil maka anda akan mendapatkan key, seperti gambar di bawah ini :



OAuth client

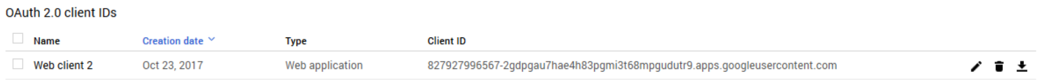
Here is your client ID

827927996567-2gdpgau7hae4h83pgmi3t68mpgudutr9.apps.googleusercontent.co

Here is your client secret

gIqHPWmEYTGeFKkjNz-1FmJu

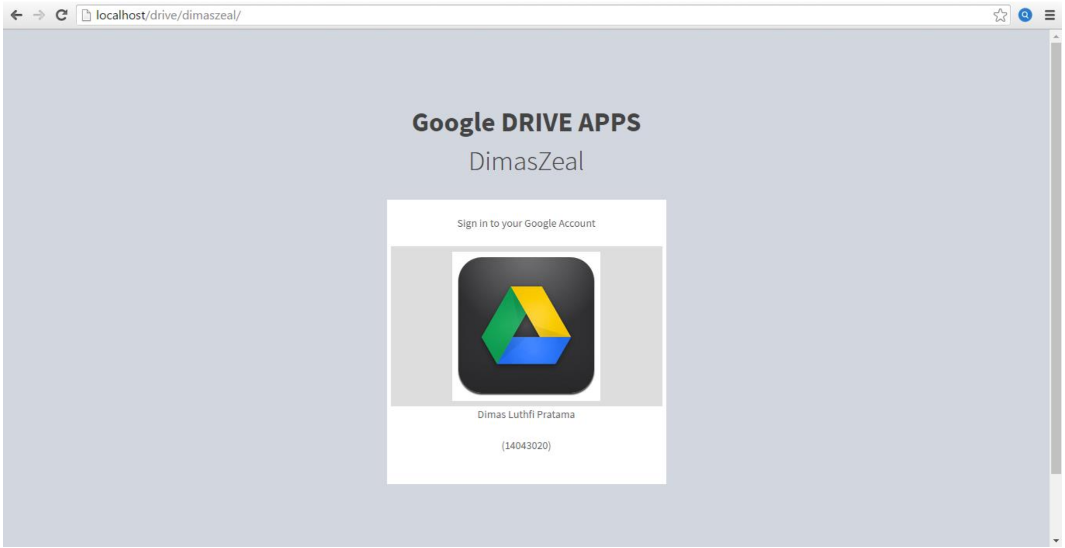
Dan jika anda ingin mendownload file json tinggal klik download seperti gambar dibawah ini:



OAuth 2.0 client IDs

□	Name	Creation date	Type	Client ID
□	Web client 2	Oct 23, 2017	Web application	827927996567-2gdpgau7hae4h83pgmi3t68mpgudutr9.apps.googleusercontent.com

Tampilan Aplikasi



localhost/drive/dimaszeal/

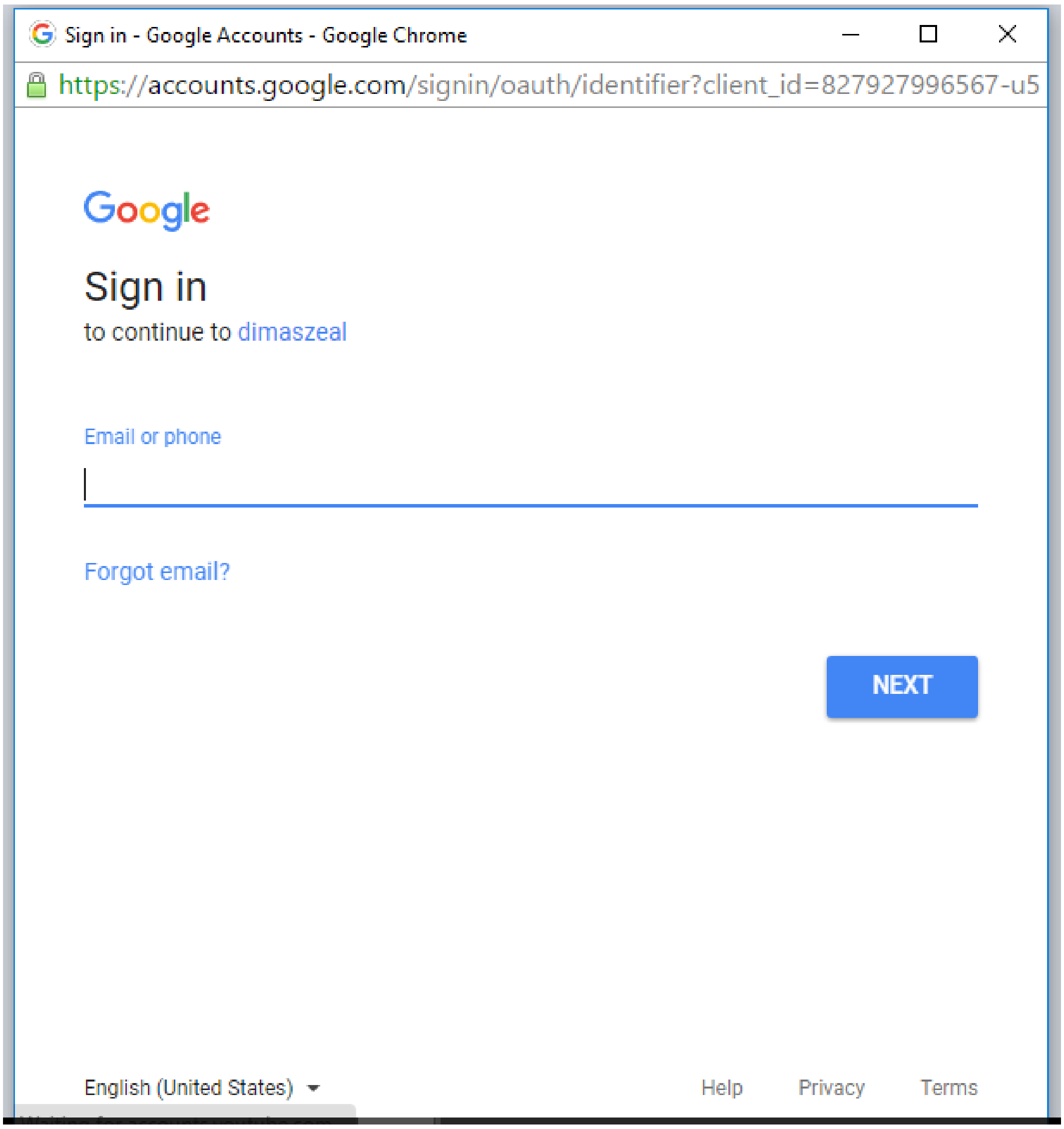
Google DRIVE APPS

DimasZeal

Sign in to your Google Account

Dimas Luthfi Pratama

(14043020)

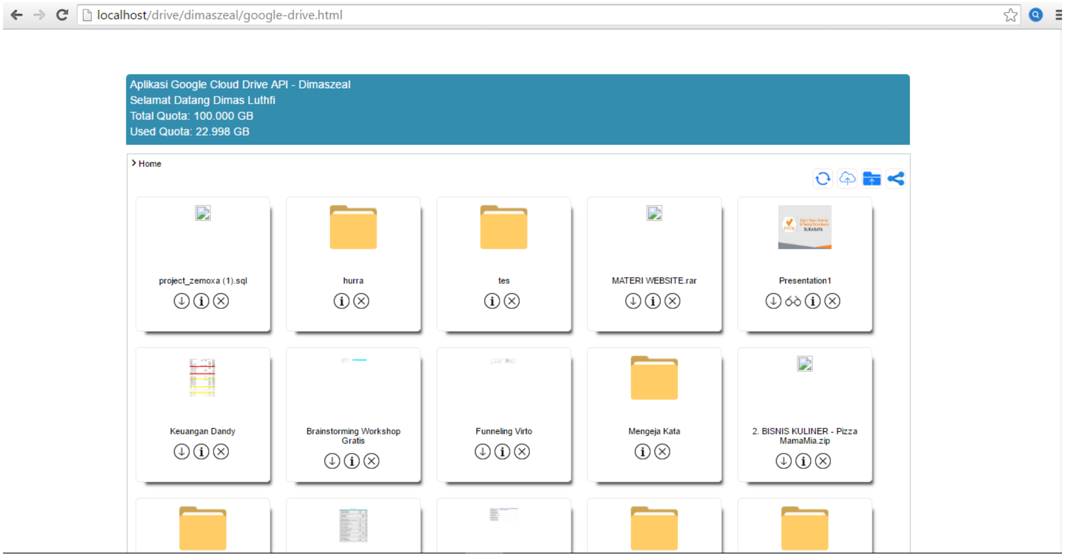


Sign in - Google Accounts - Google Chrome

https://accounts.google.com/signin/oauth/identifier?client_id=827927996567-u5

Sign in

to continue to dimaszeal

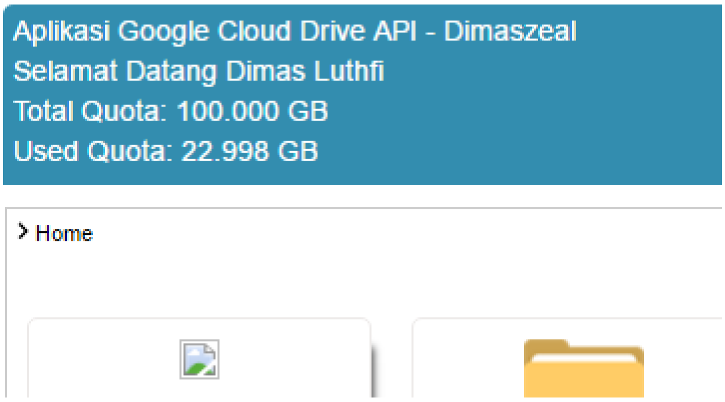


localhost/drive/dimaszeal/google-drive.html

Aplikasi Google Cloud Drive API - Dimaszeal
Selamat Datang Dimas Luthfi
Total Quota: 100.000 GB
Used Quota: 22.998 GB

Home
project_zemoxa (1).sql
hurra
tes
MATERI WEBSITE.rar
Presentation1
Keuangan Dandy
Brainstorming Workshop Gratis
Funneling Virto
Mengeja Kata
2. BISNIS KULINER - Pizza MamaMia.zip

Disini bisa menampilkan detail akun google drive anda

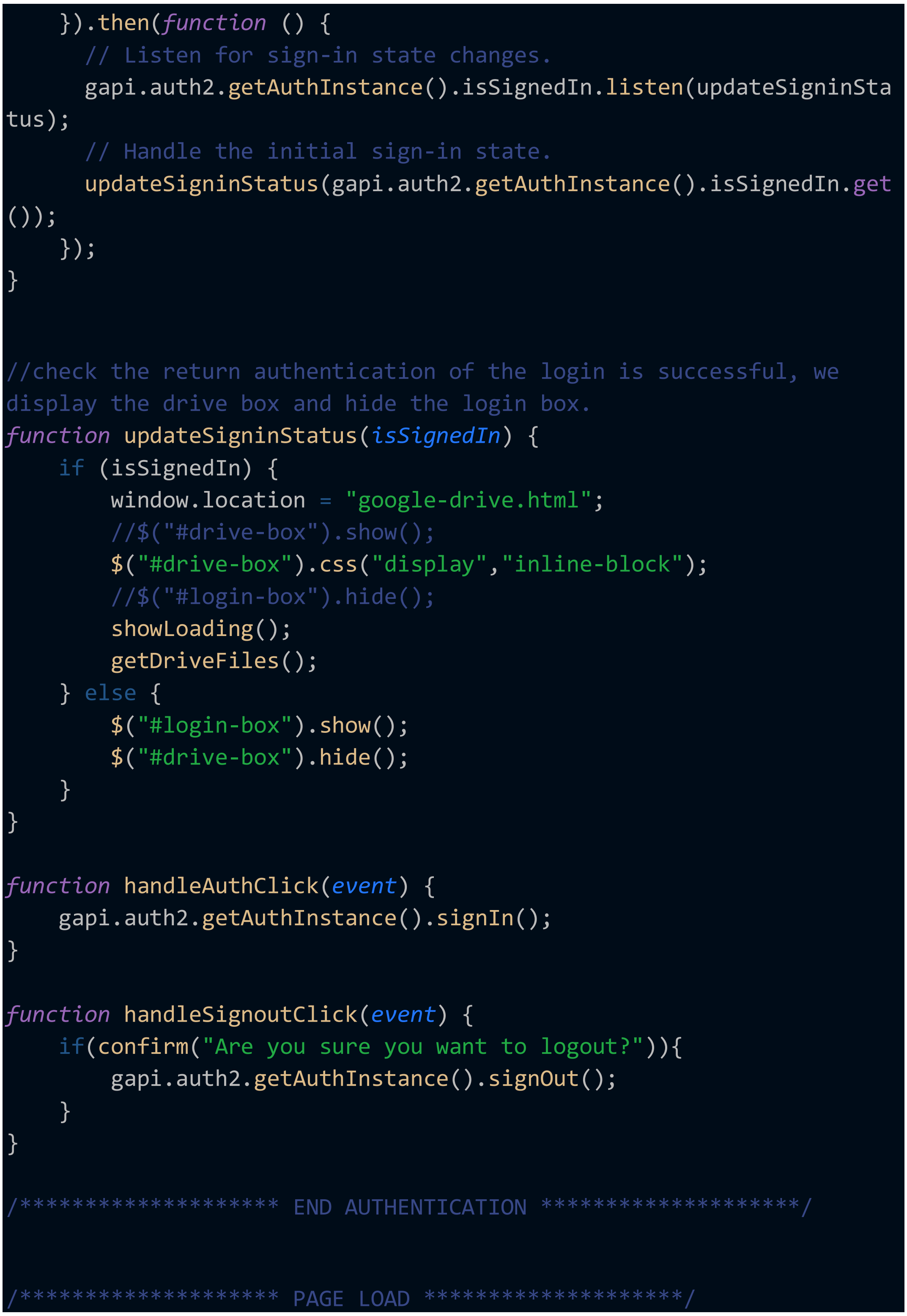


Aplikasi Google Cloud Drive API - Dimaszeal
Selamat Datang Dimas Luthfi
Total Quota: 100.000 GB
Used Quota: 22.998 GB

Source Code

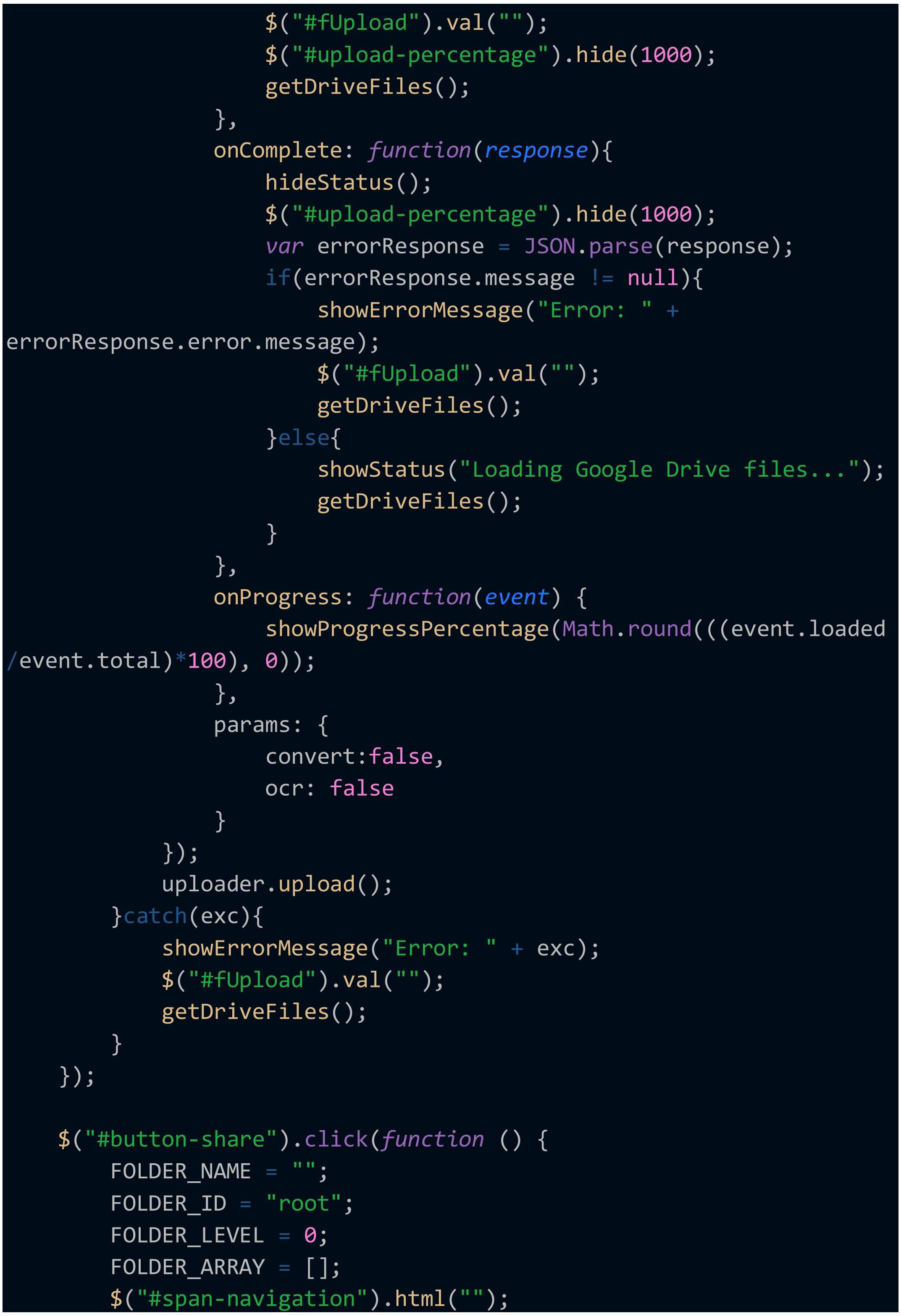
```
/*****
```

```
Bytutorial.com - online community to share articles for web and
mobile programming and designers.
Author: Andy Suwandy

NOTE: Please change the CLIENT ID by creating your own app in
google.
In order to work in your local computer, please change the client ID
in the code and set the url of where the google drive app will be
loaded.
otherwise you should get an error message saying the url you try to
load does not match.
****/

/******************** GLOBAL VARIABLES ********************/
var SCOPES = ['https://www.googleapis.com/auth/drive','profile'];
var CLIENT_ID = '827927996567-
u5p7hi20m6brr9nfa912njko9gb1k7a5.apps.googleusercontent.com';
var API_KEY = 'AIzaSyCZ0eNDy8JLCPbezzHA_sGI6BVibQohk4g';
var FOLDER_NAME = "";
var FOLDER_ID = "root";
var FOLDER_PERMISSION = true;
var FOLDER_LEVEL = 0;
var NO_OF_FILES = 1000;
var DRIVE_FILES = [];
var FILE_COUNTER = 0;
var FOLDER_ARRAY = [];

/******************** AUTHENTICATION ********************/

 function handleClientLoad() {
    // Load the API client and auth2 library
    gapi.load('client:auth2', initClient);
}

//authorize apps
 function initClient() {
    gapi.client.init({
        //apiKey: API_KEY, //THIS IS OPTIONAL AND WE DONT ACTUALLY
NEED THIS, BUT I INCLUDE THIS AS EXAMPLE
        clientId: CLIENT_ID,
        scope: SCOPES.join(' ')
```

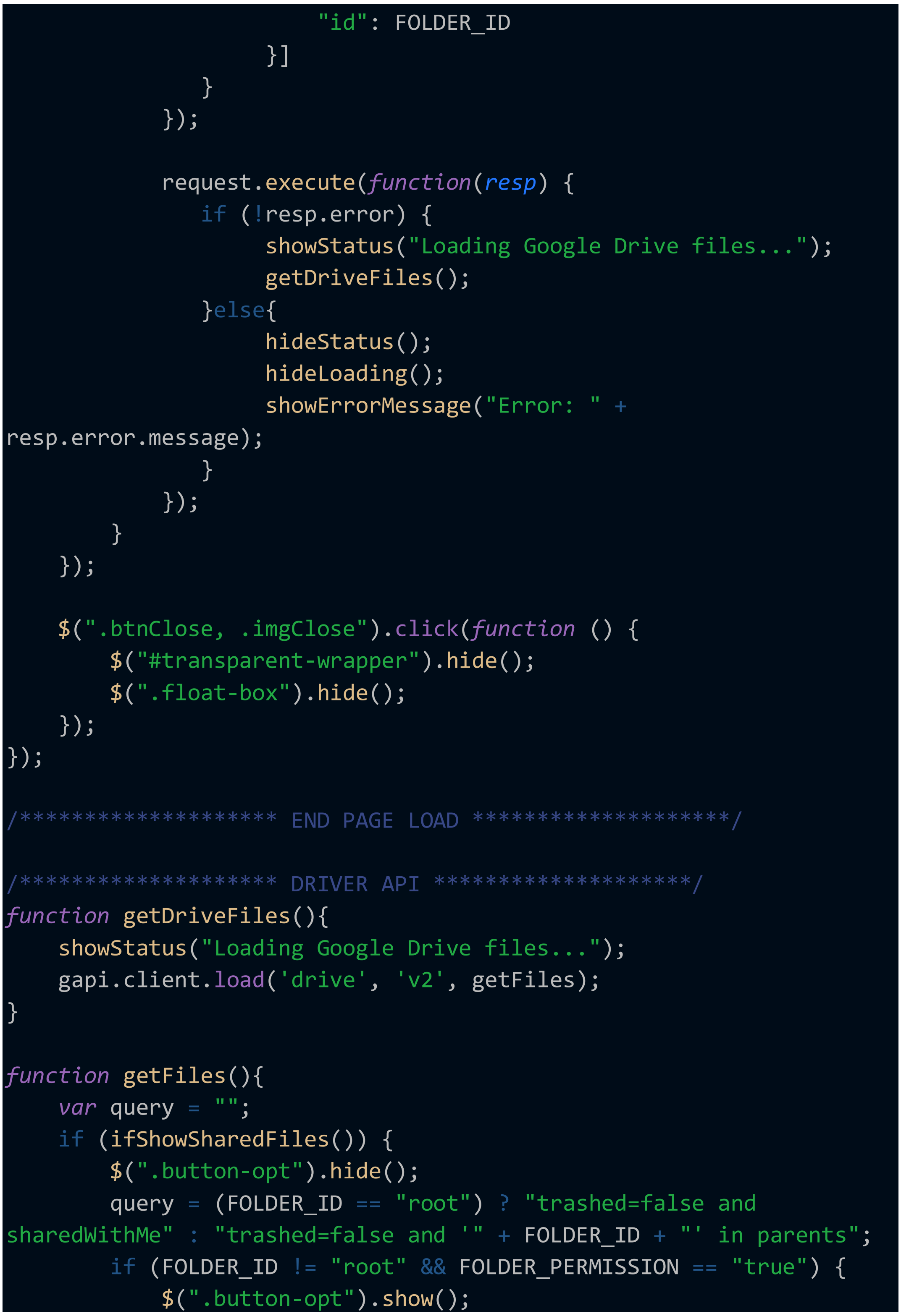


```
    }).then(function () {
      // Listen for sign-in state changes.
      gapi.auth2.getAuthInstance().isSignedIn.listen(updateSigninSta
tus);
      // Handle the initial sign-in state.
      updateSigninStatus(gapi.auth2.getAuthInstance().isSignedIn.get
());
    });
}


//check the return authentication of the login is successful, we
display the drive box and hide the login box.
function updateSigninStatus(isSignedIn) {
    if (isSignedIn) {
        window.location = "google-drive.html";
        //$("#drive-box").show();
        $("#drive-box").css("display","inline-block");
        //$("#login-box").hide();
        showLoading();
        getDriveFiles();
    } else {
        $("#login-box").show();
        $("#drive-box").hide();
    }
}

function handleAuthClick(event) {
    gapi.auth2.getAuthInstance().signIn();
}

function handleSignoutClick(event) {
    if(confirm("Are you sure you want to logout?")){
        gapi.auth2.getAuthInstance().signOut();
    }
}

/******************** END AUTHENTICATION ********************/


/******************** PAGE LOAD ********************/
```

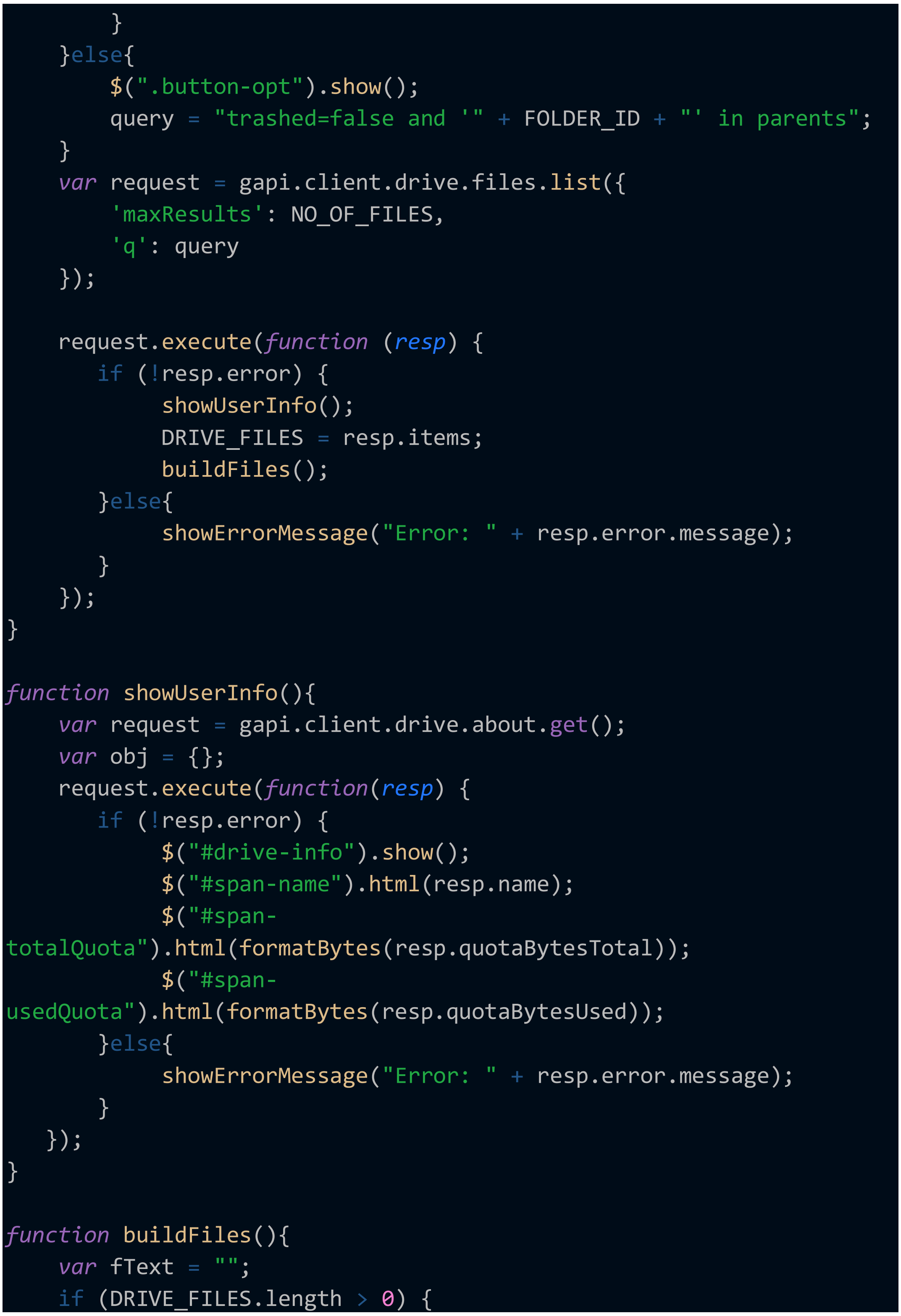
```
$(function(){
    $("#button-reload").click(function () {
        showLoading();
        showStatus("Loading Google Drive files...");
        getDriveFiles();
    });

    $("#button-upload").click(function () {
        $("#fUpload").click();
    });

     $("#fUpload").bind("change", function () {
        var uploadObj = $("[id$=fUpload]");
        showLoading();
        showStatus("Uploading file in progress...");
        var file = uploadObj.prop("files")[0];
        var metadata = {
            'title': file.name,
            'description': "bytutorial.com File Upload",
            'mimeType': file.type || 'application/octet-stream',
            "parents": [{
                "kind": "drive#file",
                "id": FOLDER_ID
            }]
        };
        var arrayBufferView = new Uint8Array(file);
        var uploadData = new Blob(arrayBufferView, {type: file.type
|| 'application/octet-stream'});
        showProgressPercentage(0);

        try{
            var uploader =new MediaUploader({
                file: file,
                token:
gapi.auth2.getAuthInstance().currentUser.get().getAuthResponse().acc
ess_token,
                metadata: metadata,
                onError: function(response){
                    var errorResponse = JSON.parse(response);
                    showErrorMessage("Error: " +
errorResponse.error.message);
```



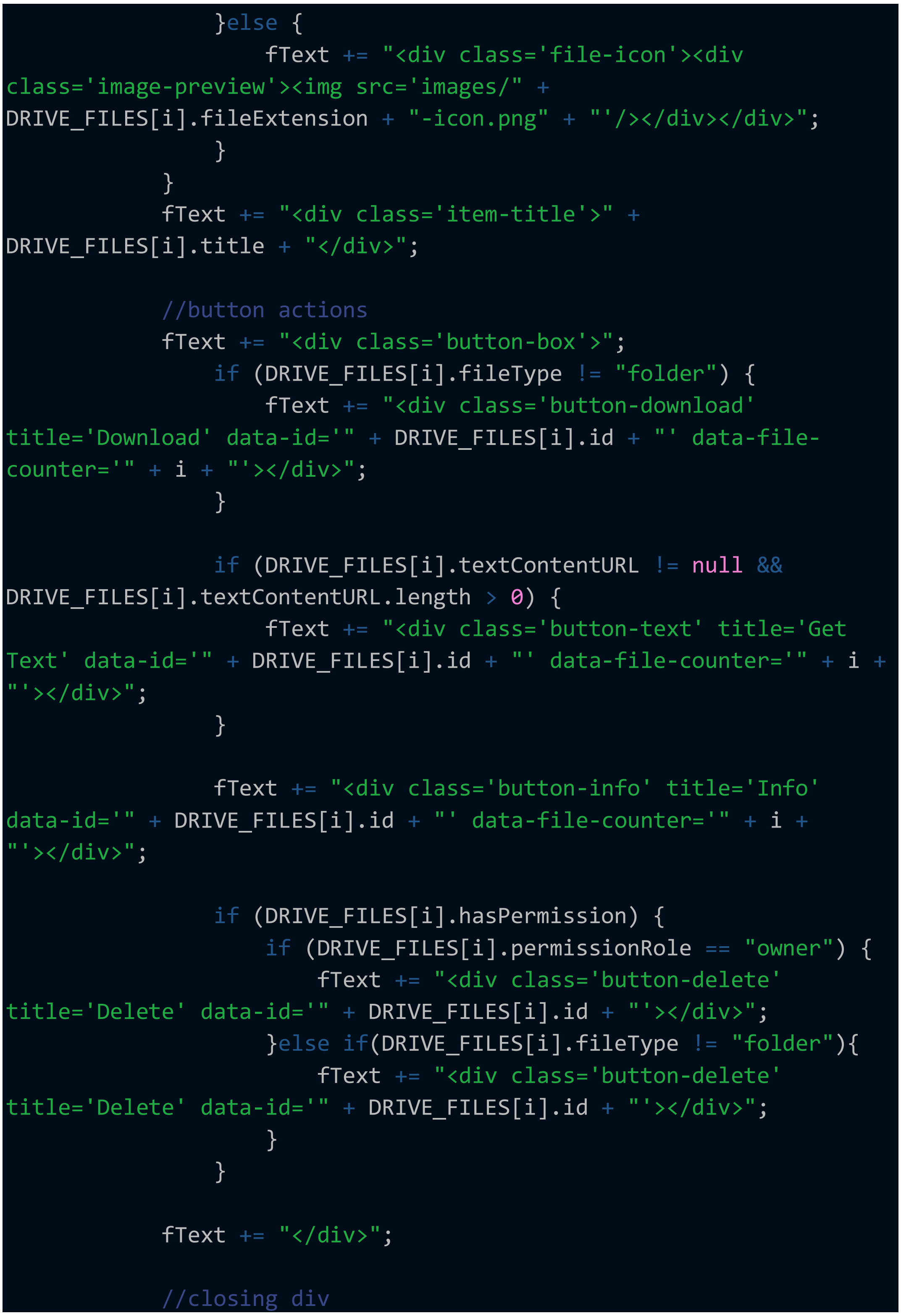
```
                    $("#fUpload").val("");
                    $("#upload-percentage").hide(1000);
                    getDriveFiles();
                },
                onComplete: function(response){
                    hideStatus();
                    $("#upload-percentage").hide(1000);
                    var errorResponse = JSON.parse(response);
                    if(errorResponse.message != null){
                        showErrorMessage("Error: " +
errorResponse.error.message);
                        $("#fUpload").val("");
                        getDriveFiles();
                    }else{
                        showStatus("Loading Google Drive files...");
                        getDriveFiles();
                    }
                },
                onProgress: function(event) {
                    showProgressPercentage(Math.round(((event.loaded
/event.total)*100), 0));
                },
                params: {
                    convert:false,
                    ocr: false
                }
            });
            uploader.upload();
        }catch(exc){
            showErrorMessage("Error: " + exc);
            $("#fUpload").val("");
            getDriveFiles();
        }
    });

    $("#button-share").click(function () {
        FOLDER_NAME = "";
        FOLDER_ID = "root";
        FOLDER_LEVEL = 0;
        FOLDER_ARRAY = [];
        $("#span-navigation").html("");
        $(this).toggleClass("flash");
        if($(this).attr("class").indexOf("flash") >= 0){
            $("#span-sharemode").html("ON");
        }else{
            $("#span-sharemode").html("OFF");
        }
        showLoading();
        showStatus("Loading Google Drive files...");
        getDriveFiles();
    });

    $("#button-addfolder").click(function () {
        $("#transparent-wrapper").show();
        $("#float-box").show();
        $("#txtFolder").val("");
    });

    $("#btnAddFolder").click(function () {
        if ($("#txtFolder").val() == "") {
            alert("Please enter the folder name");
        } else {
            $("#transparent-wrapper").hide();
            $("#float-box").hide();
            showLoading();
            showStatus("Creating folder in progress...");
            var access_token =
gapi.auth2.getAuthInstance().currentUser.get().getAuthResponse().acc
ess_token;
            var request = gapi.client.request({
               'path': '/drive/v2/files/',
               'method': 'POST',
               'headers': {
                   'Content-Type': 'application/json',
                   'Authorization': 'Bearer ' + access_token,
               },
               'body':{
                   "title" : $("#txtFolder").val(),
                   "mimeType" : "application/vnd.google-
apps.folder",
                   "parents": [{
                        "kind": "drive#file",
```



```
                        "id": FOLDER_ID
                    }]
               }
            });

            request.execute(function(resp) {
               if (!resp.error) {
                    showStatus("Loading Google Drive files...");
                    getDriveFiles();
               }else{
                    hideStatus();
                    hideLoading();
                    showErrorMessage("Error: " +
resp.error.message);
               }
            });
        }
    });

    $(".btnClose, .imgClose").click(function () {
        $("#transparent-wrapper").hide();
        $(".float-box").hide();
    });
});

/******************** END PAGE LOAD ********************/

/******************** DRIVER API ********************/
function getDriveFiles(){
    showStatus("Loading Google Drive files...");
    gapi.client.load('drive', 'v2', getFiles);
}

function getFiles(){
    var query = "";
    if (ifShowSharedFiles()) {
        $(".button-opt").hide();
        query = (FOLDER_ID == "root") ? "trashed=false and
sharedWithMe" : "trashed=false and '" + FOLDER_ID + "' in parents";
        if (FOLDER_ID != "root" && FOLDER_PERMISSION == "true") {
            $(".button-opt").show();
```



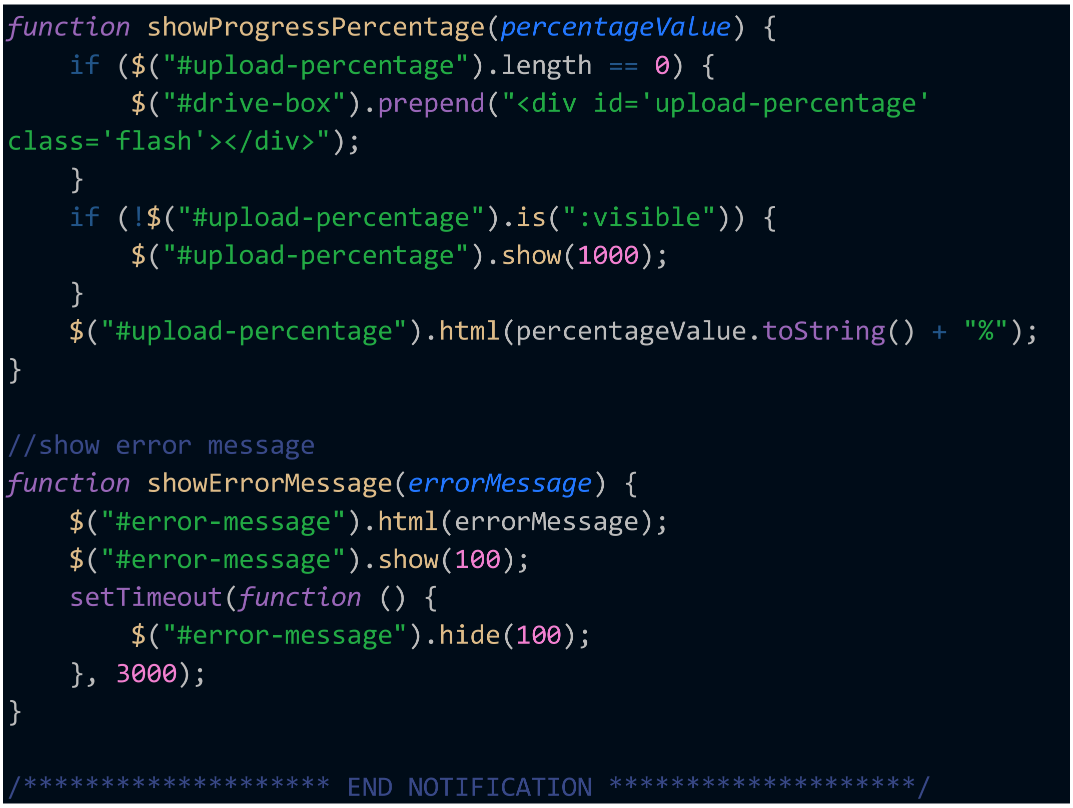
```
        }
    }else{
        $(".button-opt").show();
        query = "trashed=false and '" + FOLDER_ID + "' in parents";
    }
    var request = gapi.client.drive.files.list({
        'maxResults': NO_OF_FILES,
        'q': query
    });

    request.execute(function (resp) {
       if (!resp.error) {
            showUserInfo();
            DRIVE_FILES = resp.items;
            buildFiles();
       }else{
            showErrorMessage("Error: " + resp.error.message);
       }
    });
}

function showUserInfo(){
    var request = gapi.client.drive.about.get();
    var obj = {};
    request.execute(function(resp) {
       if (!resp.error) {
            $("#drive-info").show();
            $("#span-name").html(resp.name);
            $("#span-
totalQuota").html(formatBytes(resp.quotaBytesTotal));
            $("#span-
usedQuota").html(formatBytes(resp.quotaBytesUsed));
       }else{
            showErrorMessage("Error: " + resp.error.message);
       }
   });
}

function buildFiles(){
    var fText = "";
    if (DRIVE_FILES.length > 0) {
        for (var i = 0; i < DRIVE_FILES.length; i++) {
            DRIVE_FILES[i].textContentURL = "";
            DRIVE_FILES[i].level = (parseInt(FOLDER_LEVEL) +
1).toString();
            DRIVE_FILES[i].parentID = (DRIVE_FILES[i].parents.length
> 0) ? DRIVE_FILES[i].parents[0].id : "";
            DRIVE_FILES[i].thumbnailLink =
DRIVE_FILES[i].thumbnailLink || '';
            DRIVE_FILES[i].fileType =  (DRIVE_FILES[i].fileExtension
== null) ? "folder" : "file";
            DRIVE_FILES[i].permissionRole =
DRIVE_FILES[i].userPermission.role;
            DRIVE_FILES[i].hasPermission =
(DRIVE_FILES[i].permissionRole == "owner" ||
DRIVE_FILES[i].permissionRole == "writer");
            var textContentURL = '';
            if(DRIVE_FILES[i]['exportLinks'] != null){
                DRIVE_FILES[i].fileType = "file";
                DRIVE_FILES[i].textContentURL =
DRIVE_FILES[i]['exportLinks']['text/plain'];
            }
            var textTitle = (DRIVE_FILES[i].fileType != "file") ?
"Browse " + DRIVE_FILES[i].title : DRIVE_FILES[i].title;

            fText += "<div class='" + DRIVE_FILES[i].fileType + "-
box'>";
            if (DRIVE_FILES[i].fileType != "file") {
                fText += "<div class='folder-icon' data-level='" +
DRIVE_FILES[i].level + "' data-parent='" + DRIVE_FILES[i].parentID +
"' data-size='" + DRIVE_FILES[i].fileSize + "' data-id='" +
DRIVE_FILES[i].id + "' title='" + textTitle + "' data-name='" +
DRIVE_FILES[i].title + "' data-has-permission='"
+DRIVE_FILES[i].hasPermission + "'><div class='image-preview'><img
src='images/folder.png'/></div></div>";
            } else {
                if (DRIVE_FILES[i].thumbnailLink) {
                    fText += "<div class='image-icon'><div
class='image-preview'><a href='" +
DRIVE_FILES[i].thumbnailLink.replace("s220", "s800") + "' data-
lightbox='image-" + i + "'><img src='" +
DRIVE_FILES[i].thumbnailLink + "'/></a></div></div>";
```



```
        }else {
          fText += "<div class='file-icon'><div
class='image-preview'><img src='images/" +
DRIVE_FILES[i].fileExtension + "-icon.png" + "'/></div></div>";
        }
      }
      fText += "<div class='item-title'>" +
DRIVE_FILES[i].title + "</div>";

      //button actions
      fText += "<div class='button-box'>";
        if (DRIVE_FILES[i].fileType != "folder") {
          fText += "<div class='button-download'
title='Download' data-id='" + DRIVE_FILES[i].id + "' data-file-
counter='" + i + "'></div>";
        }

        if (DRIVE_FILES[i].textContentURL != null &&
DRIVE_FILES[i].textContentURL.length > 0) {
          fText += "<div class='button-text' title='Get
Text' data-id='" + DRIVE_FILES[i].id + "' data-file-counter='" + i +
"'></div>";
        }

        fText += "<div class='button-info' title='Info'
data-id='" + DRIVE_FILES[i].id + "' data-file-counter='" + i +
"'></div>";

        if (DRIVE_FILES[i].hasPermission) {
          if (DRIVE_FILES[i].permissionRole == "owner") {
            fText += "<div class='button-delete'
title='Delete' data-id='" + DRIVE_FILES[i].id + "'></div>";
          }else if(DRIVE_FILES[i].fileType != "folder"){
            fText += "<div class='button-delete'
title='Delete' data-id='" + DRIVE_FILES[i].id + "'></div>";
          }
        }

      fText += "</div>";

      //closing div
```

```
            fText += "</div>";
        }
    } else {
        fText = 'No files found.';
    }
    hideStatus();
    $("#drive-content").html(fText);
    initDriveButtons();
    hideLoading();
}

//Initialize the click button for each individual drive file/folder
//this need to be recalled everytime the Google Drive data is
generated
function initDriveButtons(){
    //Initiate the delete button click
    $(".button-delete").unbind("click");
    $(".button-delete").click(function () {
        var c = confirm("Are you sure you want to delete this?");
        if (c) {
            showLoading();
            showStatus("Deleting file in progress...");
            var request = gapi.client.drive.files.delete({
                'fileId': $(this).attr("data-id")
            });

            request.execute(function(resp) {
               hideStatus();
               if (resp.error) {
                    showErrorMessage("Error: " +
resp.error.message);
               }
               getDriveFiles();
            });
        }
    });

    //Initiate the download button
    $(".button-download").unbind("click");
    $(".button-download").click(function () {
        showLoading();
        showStatus("Downloading file in progress...");
        FILE_COUNTER = $(this).attr("data-file-counter");
        setTimeout(function () {
            //If there is a text version, we get this version
instead.
            if(DRIVE_FILES[FILE_COUNTER].webContentLink == null){
                window.open(DRIVE_FILES[FILE_COUNTER]['exportLinks']
['text/plain']);
            }else{
                window.open(DRIVE_FILES[FILE_COUNTER].webContentLink
);
            }
            hideLoading();
            hideStatus();
        }, 1000);
    });

    $(".button-info").unbind("click");
    $(".button-info").click(function () {
        FILE_COUNTER = $(this).attr("data-file-counter");
        $("#transparent-wrapper").show();
        $("#float-box-info").show();
        if (DRIVE_FILES[FILE_COUNTER] != null) {
            var createdDate = new
Date(DRIVE_FILES[FILE_COUNTER].createdDate);
            var modifiedDate = new
Date(DRIVE_FILES[FILE_COUNTER].modifiedDate);
            $("#spanCreatedDate").html(createdDate.toString("dddd, d
MMMM yyyy h:mm:ss tt"));
            $("#spanModifiedDate").html(modifiedDate.toString("dddd,
d MMMM yyyy h:mm:ss tt"));

$("#spanOwner").html((DRIVE_FILES[FILE_COUNTER].owners[0].displayNam
e.length > 0) ? DRIVE_FILES[FILE_COUNTER].owners[0].displayName :
"");
            $("#spanTitle").html(DRIVE_FILES[FILE_COUNTER].title);
            $("#spanSize").html((DRIVE_FILES[FILE_COUNTER].fileSize
== null) ? "N/A" : formatBytes(DRIVE_FILES[FILE_COUNTER].fileSize));

$("#spanExtension").html(DRIVE_FILES[FILE_COUNTER].fileExtension);
        }
    });

    //Initiate the get text button
    $(".button-text").unbind("click");
    $(".button-text").click(function () {
        FILE_COUNTER = $(this).attr("data-file-counter");
        showLoading();
        showStatus("Getting text file in progress...");
        var accessToken =
gapi.auth2.getAuthInstance().currentUser.get().getAuthResponse().acc
ess_token;
        var xhr = new XMLHttpRequest();
        xhr.open('GET',
DRIVE_FILES[FILE_COUNTER]['exportLinks']['text/plain']);
        xhr.setRequestHeader('Authorization', 'Bearer ' +
accessToken);
        xhr.onload = function() {
            callBackGetText(xhr.responseText);
        };
        xhr.onerror = function() {
            callBackGetText(null);
        };
        xhr.send();
    });

    //Initiate the click folder browse icon
    $(".folder-icon").unbind("click");
    $(".folder-icon").click(function () {
        browseFolder($(this));
    });

    //Initiate the breadcrumb navigation link click
    $("#drive-breadcrumb a").unbind("click");
    $("#drive-breadcrumb a").click(function () {
        browseFolder($(this));
    });
}


//browse folder
function browseFolder(obj) {
    FOLDER_ID = $(obj).attr("data-id");
    FOLDER_NAME = $(obj).attr("data-name");
    FOLDER_LEVEL = parseInt($(obj).attr("data-level"));
    FOLDER_PERMISSION = $(obj).attr("data-has-permission");

    if (typeof FOLDER_NAME === "undefined") {
        FOLDER_NAME = "";
        FOLDER_ID = "root";
        FOLDER_LEVEL = 0;
        FOLDER_PERMISSION = true;
        FOLDER_ARRAY = [];
    } else {
        if (FOLDER_LEVEL == FOLDER_ARRAY.length && FOLDER_LEVEL > 0)
{
            //do nothing
        } else if (FOLDER_LEVEL < FOLDER_ARRAY.length) {
            var tmpArray = cloneObject(FOLDER_ARRAY);
            FOLDER_ARRAY = [];

            for (var i = 0; i < tmpArray.length; i++) {
                FOLDER_ARRAY.push(tmpArray[i]);
                if (tmpArray[i].Level >= FOLDER_LEVEL) { break; }
            }
        } else {
            var fd = {
                Name: FOLDER_NAME,
                ID: FOLDER_ID,
                Level: FOLDER_LEVEL,
                Permission: FOLDER_PERMISSION
            }
            FOLDER_ARRAY.push(fd);
        }
    }

    var sbNav = "";
    for (var i = 0; i < FOLDER_ARRAY.length; i++) {
        sbNav +="<span class='breadcrumb-arrow'></span>";
        sbNav +="<span class='folder-name'><a data-id='" +
FOLDER_ARRAY[i].ID + "' data-level='" + FOLDER_ARRAY[i].Level + "'
data-name='" + FOLDER_ARRAY[i].Name + "' data-has-permission='" +
FOLDER_PERMISSION + "'>" + FOLDER_ARRAY[i].Name + "</a></span>";
```

```
    }
    $("#span-navigation").html(sbNav.toString());

    showLoading();
    showStatus("Loading Google Drive files...");
    getDriveFiles();
}

//call back function for getting text
function callBackGetText(response){
    if(response == null){
         showErrorMessage("Error getting text content.");
    }else{
        hideLoading();
        hideStatus();
        $("#transparent-wrapper").show();
        $("#float-box-text").show();
        $("#text-content").html(response.replace(/(\r\n|\n|\r)/gm,
"<br>"));
    }
}

//function to clone an object
function cloneObject(obj) {
    if (obj === null || typeof obj !== 'object') {
        return obj;
    }
    var temp = obj.constructor();
    for (var key in obj) {
        temp[key] = cloneObject(obj[key]);
    }
    return temp;
}

//show whether the display mode is share files or not
function ifShowSharedFiles() {
    return ($("#button-share.flash").length > 0) ? true : false;
}

//function to return bytes into different string data format
function formatBytes(bytes) {
    if (bytes < 1024) return bytes + " Bytes";
    else if (bytes < 1048576) return (bytes / 1024).toFixed(3) + "
KB";
    else if (bytes < 1073741824) return (bytes / 1048576).toFixed(3)
+ " MB";
    else return (bytes / 1073741824).toFixed(3) + " GB";
};

/******************** END DRIVER API ********************/



/******************** NOTIFICATION ********************/
//show loading animation
function showLoading() {
    if ($("#drive-box-loading").length === 0) {
        $("#drive-box").prepend("<div id='drive-box-
loading'></div>");
    }
    $("#drive-box-loading").html("<div id='loading-wrapper'><div
id='loading'><img src='images/loading-bubble.gif'></div></div>");
}

//hide loading animation
function hideLoading() {
    $("#drive-box-loading").html("");
}

//show status message
function showStatus(text) {
    $("#status-message").show();
    $("#status-message").html(text);
}

//hide status message
function hideStatus() {
    $("#status-message").hide();
    $("#status-message").html("");
}

//show upload progress
```



```
function showProgressPercentage(percentageValue) {
    if ($("#upload-percentage").length == 0) {
        $("#drive-box").prepend("<div id='upload-percentage'
class='flash'></div>");
    }
    if (!$("#upload-percentage").is(":visible")) {
        $("#upload-percentage").show(1000);
    }
    $("#upload-percentage").html(percentageValue.toString() + "%");
}

//show error message
function showErrorMessage(errorMessage) {
    $("#error-message").html(errorMessage);
    $("#error-message").show(100);
    setTimeout(function () {
        $("#error-message").hide(100);
    }, 3000);
}

/******************** END NOTIFICATION ********************/
```