**OAuth 2.0 framework Authorization Process (May 2020)**

**H M P K Ranjan Kumara Herath**

**Faculty of Computing**

**Sri Lanka Institute of information Technology**

**Malabe, Sri Lanka**

**MS20911058@my.sliit.lk**

**Abstract –** **In this paper emphasized that security of Information Technology of public network in aspect of security impact to a client and the resource server which is use art of the technology implementation of this era called Oauth framework 2.0. This RFC 6749 protocol standard particularly restricted the third party applications through HTTP services without authorization from the respective resource owner. In fact, it is a challenge of the expertise in this new era due to the complexity of the software industry and as well as aspect of the client requirement. A range of new sophisticated applications further create an uncertainty due to the vulnerabilities and exploitation of the threats unexpectedly. In this review is further illustrated the Oauth framework practically implemented with web applications how could compromised with Google Application Program Interface.**

**Key Words: Oauth 2.0 Framework, API, Resource Server, Resource Owner, Client, Authorization Server. Access Token, Authorization Request, Authorization Granted**

**I. INTRODUCTION**

Information security is based on very primitive but potential important three fundamental principles of security aspects such as confidentiality, integrity, and availability. They are the pillars of the information. Then the challenge would be implementation of the secure systems to prevent or mitigate such intentional misuse of the public or private properties.

Open Authorization (OAuth) is a protocol which is used to protected the data shared in a public platform such a as Google Drive API. Internet Engineering Task Force has been implemented a framework such as RFC 6749 by defining the standards protocol flow with proper steps of procedures.

The OAuth framework is illustrated mainly in four different services as following steps:

* Authorization requested from resource owner and directly granted to the client
* Sent the granted authorization to authentication server by requesting the token
* Issued the token by initializing the granted authorization
* Client requested the protected resources for the given token
* Resource server validate the access token, if valid grants the service

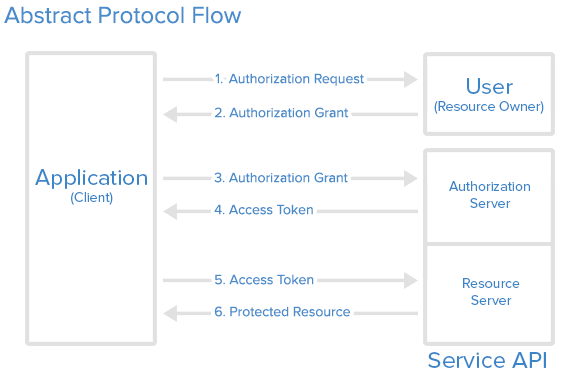


Figure 1 OAuth 2.0 Protocol Flow

1. **OAUTH ROLES:**

* Resource Owner
* Client
* Resource Server
* Authorization Server

1. **ACCESS TOKEN:**

Access token is credentials which are access protected resources. It is denoted as unidentified some encrypted date or signature in variable manner. Server can be identified both name and the password in it. Access token basically two formats such as structures and method of utilization whereas cryptographic properties.

**II. OBJECTIVES**

OAuth framework is critical for network security and quite new for many users however to be concern to protect unexpected threats exploitation of the client data. Mainly focus on here to emphasized the practically shows the application of the OAuth process in technically with created application. It is obvious that many difficulties to be faced to overcome issues and uncertainty circumstances happening such configurations and installation process. In the latter part of the review allocate the space to discussed further problems in the real issues.

**III. OAUTH 2.0 PROCESS**

There are many important terminologies to be simplified on this process for learning perspective before further evaluation.

1. **HTTP REDIRECTIONS**

Critical part of the Oath flow is the redirection the URL to another destination. After successful authorization an application the authorization server redirect the user - agent according to the access code or access token in the URL. Due to the criticality of the data in the redirected URL doesn’t redirect the user arbitrary details.

1. **APPLICATION REGISTRATION**

Application should be registered with the service before using the OAuth. This process to be done via a registration form in the “developer” or “API” of the service’s website by providing the information as application Name, Application Website and the Redirect URI or Callback URL

1. **CLIENT ID AND CLIENT SECRET**

After registration of the application, in the form of client identifier and a client secret will issue the client credentials. The client ID is publicly exposed which is used by service API identify the application and also used to build authorization URL to users.

**IV. AUTHORIZATION GRANT**

In the figure 1 shows the first steps such as cover obtaining an authorization grant and access token.

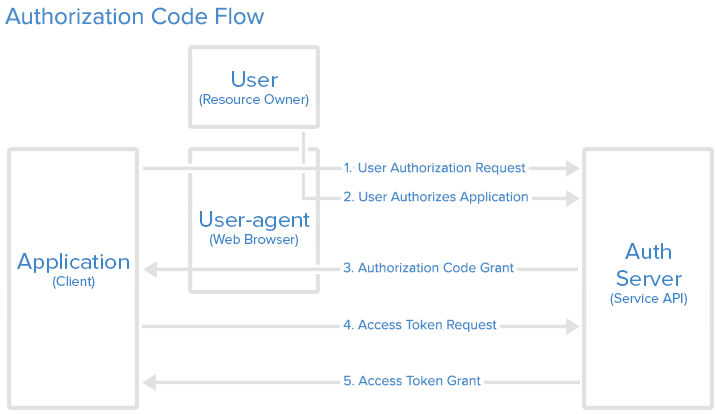
Here are the OAuth 2 authorization grant types:

* **Authorization Code**: used for server-side Applications
* **Implicit**: used for Mobile Apps or Web Applications
* **Resource Owner Password Credentials**: used for trusted Applications
* **Client Credentials**: used with Applications API access

Now we will describe grant types in more detail, their use cases and flows, in the following sections.

1. **AUTHORIZATION CODE**

For optimization of the server-side applications the **authorization code** grant type is being used apart of that it can be maintained the client Secret confidentiality. Authorization code is a process of redirection such as implication of interacting with the web browser (user-agent) and receiving API authorization codes.



**Figure 2 Authorization code flow**

# V. OAUTH 2.0 FOR CLIENT-SIDE WEB APPLICATIONS

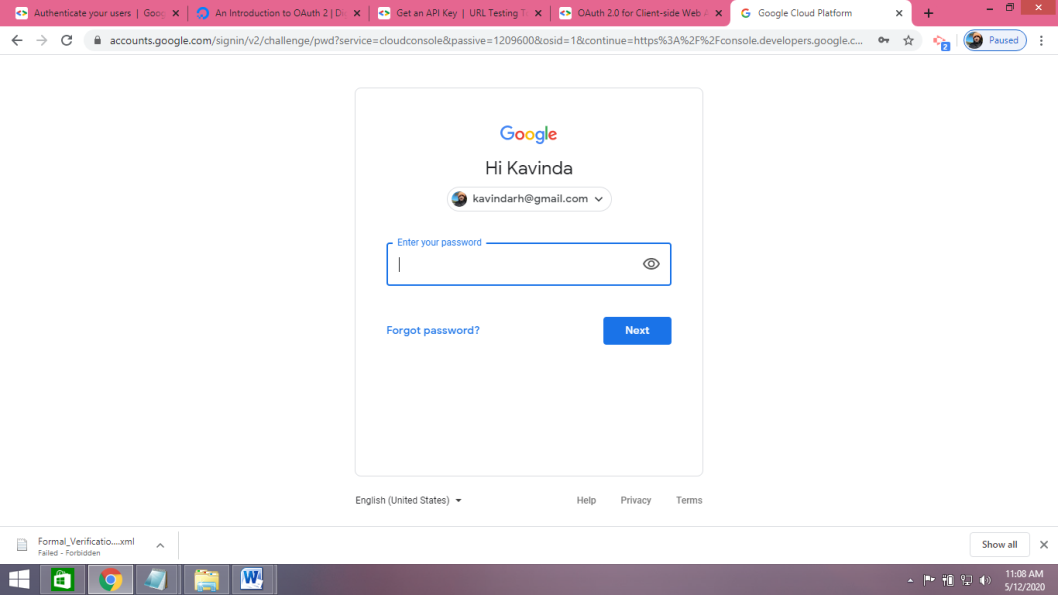
This document explain implementation of OAth 2.0 of authorization to access of Google API from JavaScript web application OAuth 2.0 is allow users to shared specific data while keeping their user name \, password and other details privately. In facts, application can store information by obtaining authorization of OAuth 2.0 in Google API.

### ENABLE APIS FOR YOUR PROJECT

* [Open the API Library](https://console.developers.google.com/apis/library) in the Google API Console.
* Create a new project.
* Selected required API product from the Google API Family.
* Click the **Enable** button
* If prompted, enable billing.
* If prompted, read and accept the API's Terms of Service.

1. **EMAIL AUTHORIZATION**

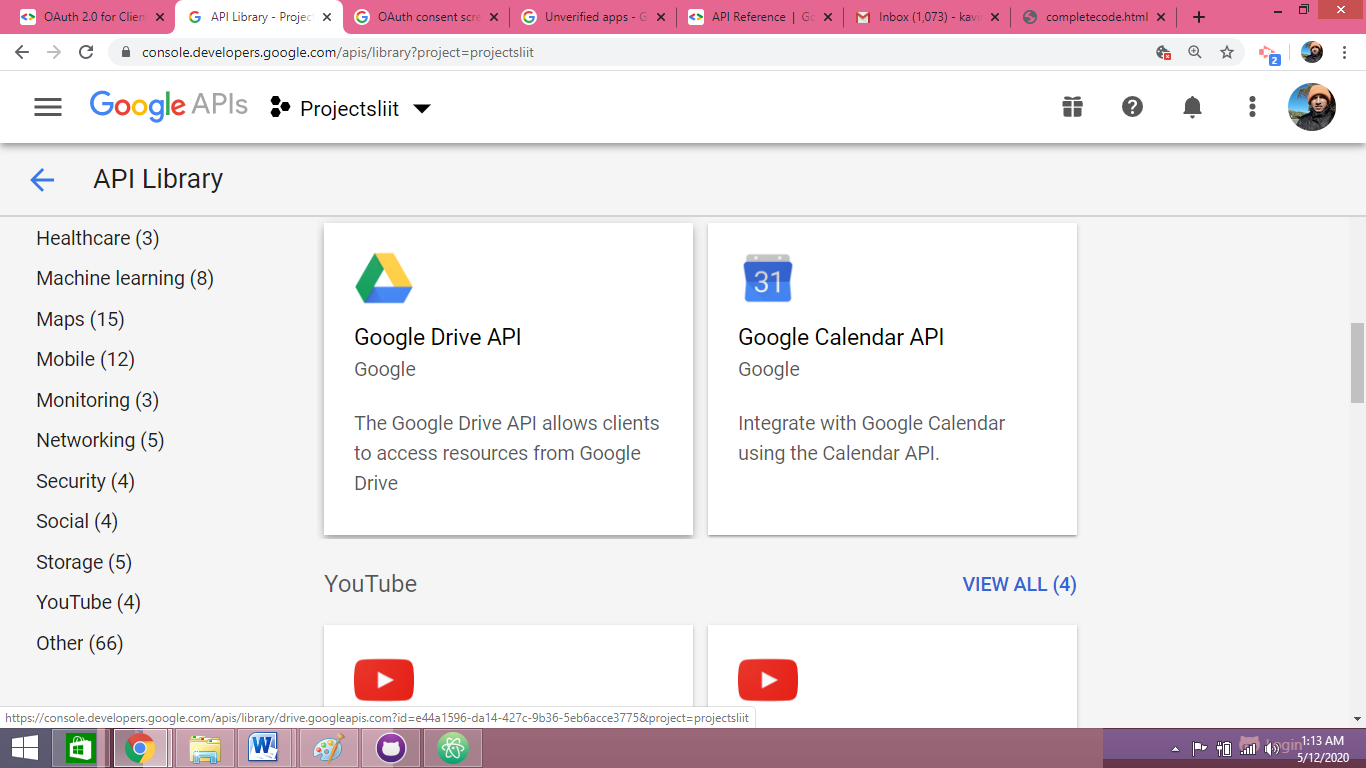
User must give the authorization as shown in the figure 3

****

**Figure 3 Authorization Link**

Click the link to get into the service then authenticate the identity then they will prompted the service authorized or deny the access. It should be Enabled the necessary application.

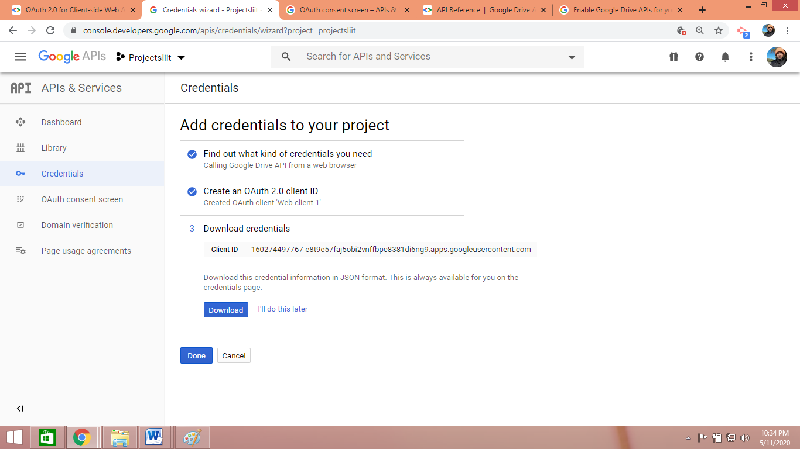
1. **USER AUTHORIZED APPLICATION**

****

**Figure 4 Enable API Google Drive**

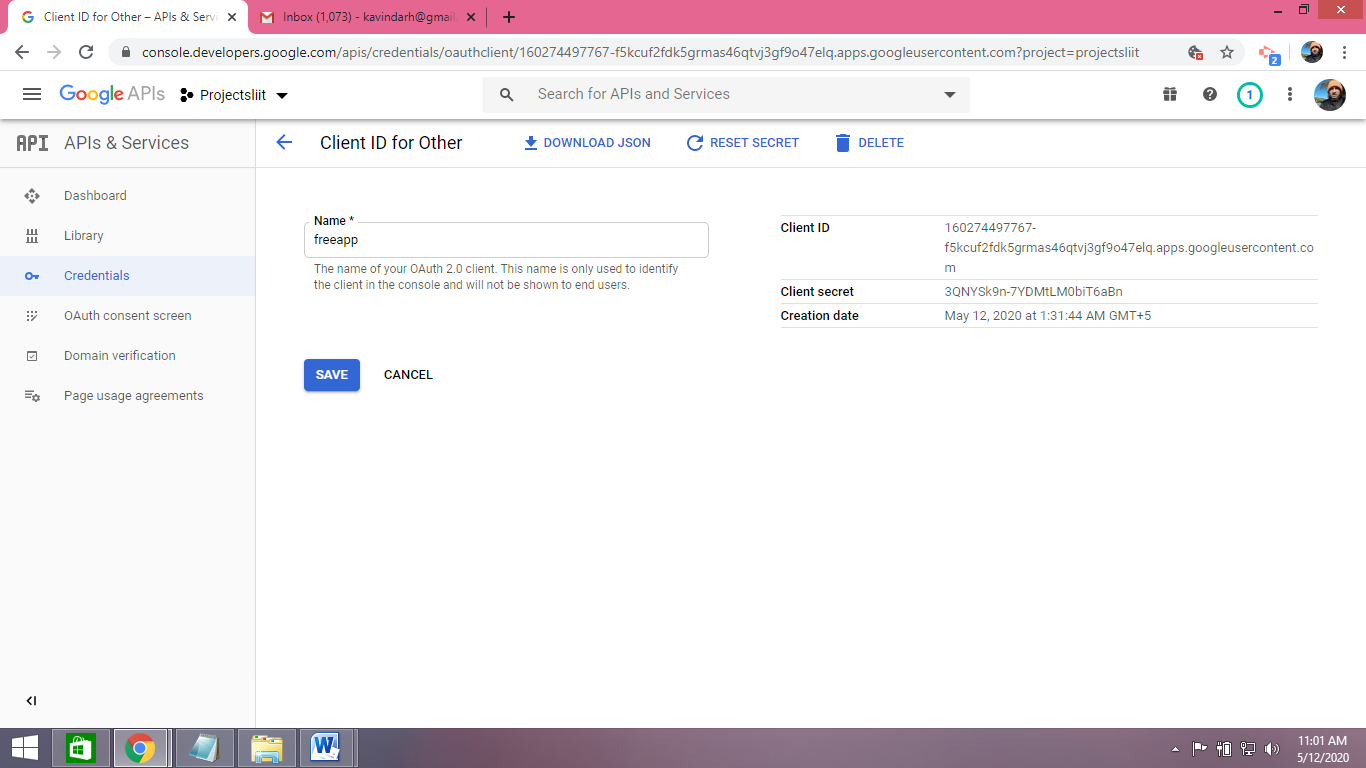
### APPLICATION RECEIVES AUTHORIZATION CODE

User must click the authorized application to redirect the user –agent applications of URL

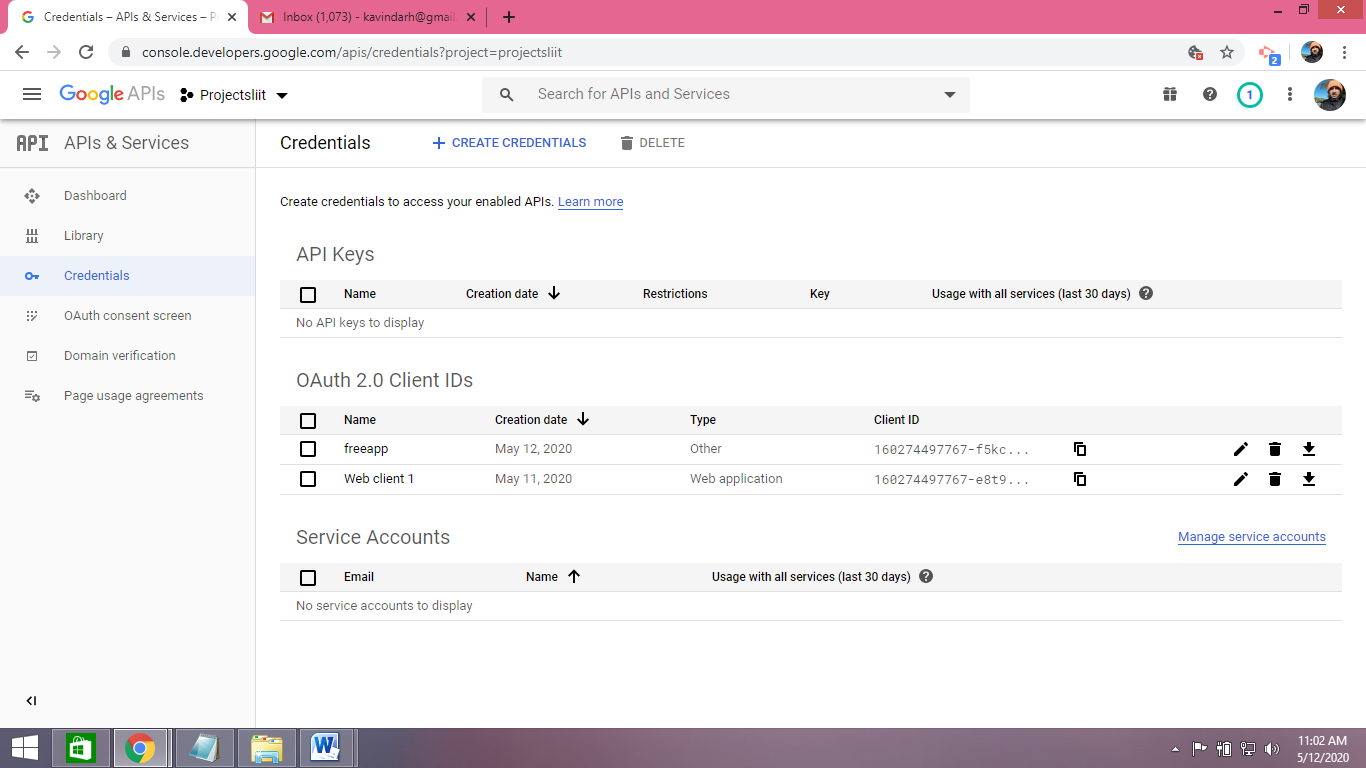


**Figure 5 Authorized Application**

### APPLICATION REQUESTS & RECEIVES ACCESS TOKEN



**Figure 6**

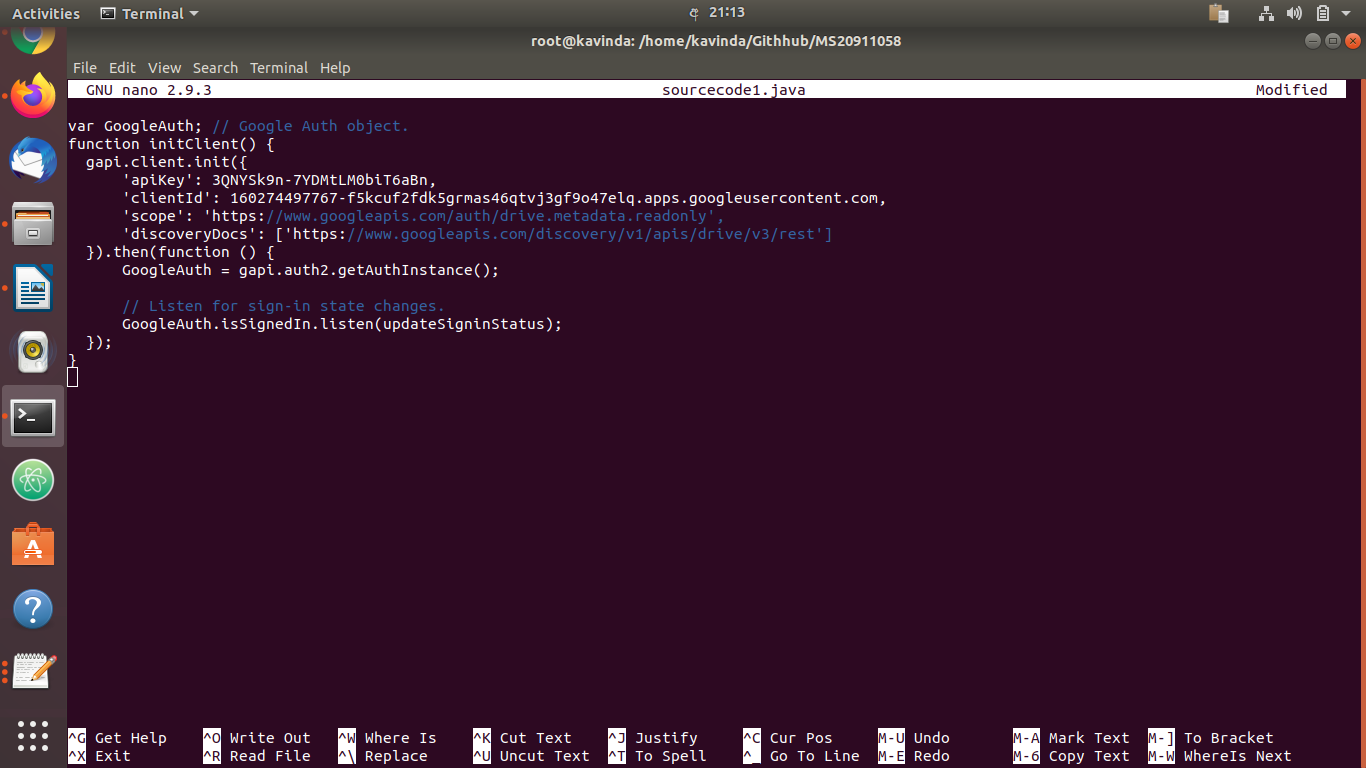
****

**Figure 7**

## VI. OBTAINING OAUTH 2.0 ACCESS TOKENS

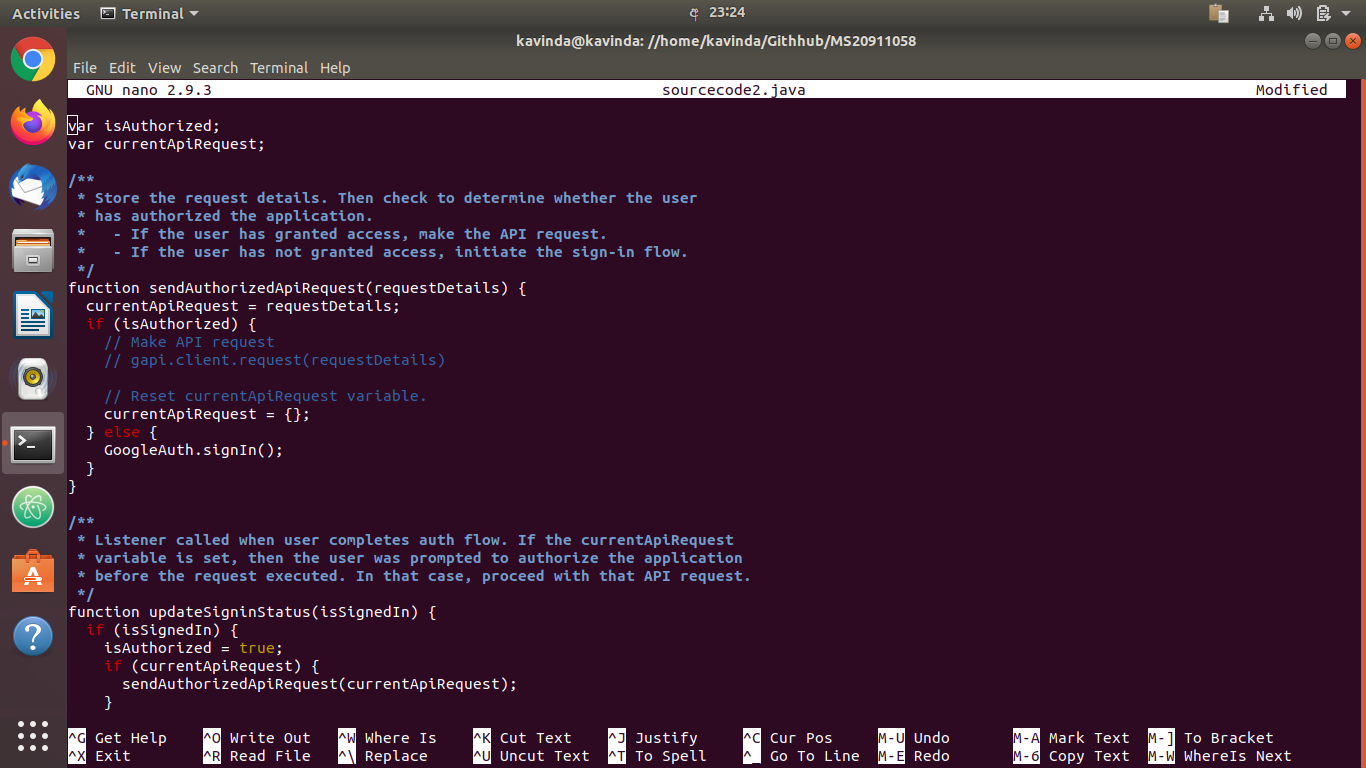
In here shows that the way of interact the user application with Google API with OAuth 2.0 obtains the consent API request from the user.

### CONFIGURE THE CLIENT OBJECT



### Figure 8 Client Object

### REDIRECT TO GOOGLE'S OAUTH 2.0 SERVE



### GOOGLE PROMPTS USER FOR CONSENT

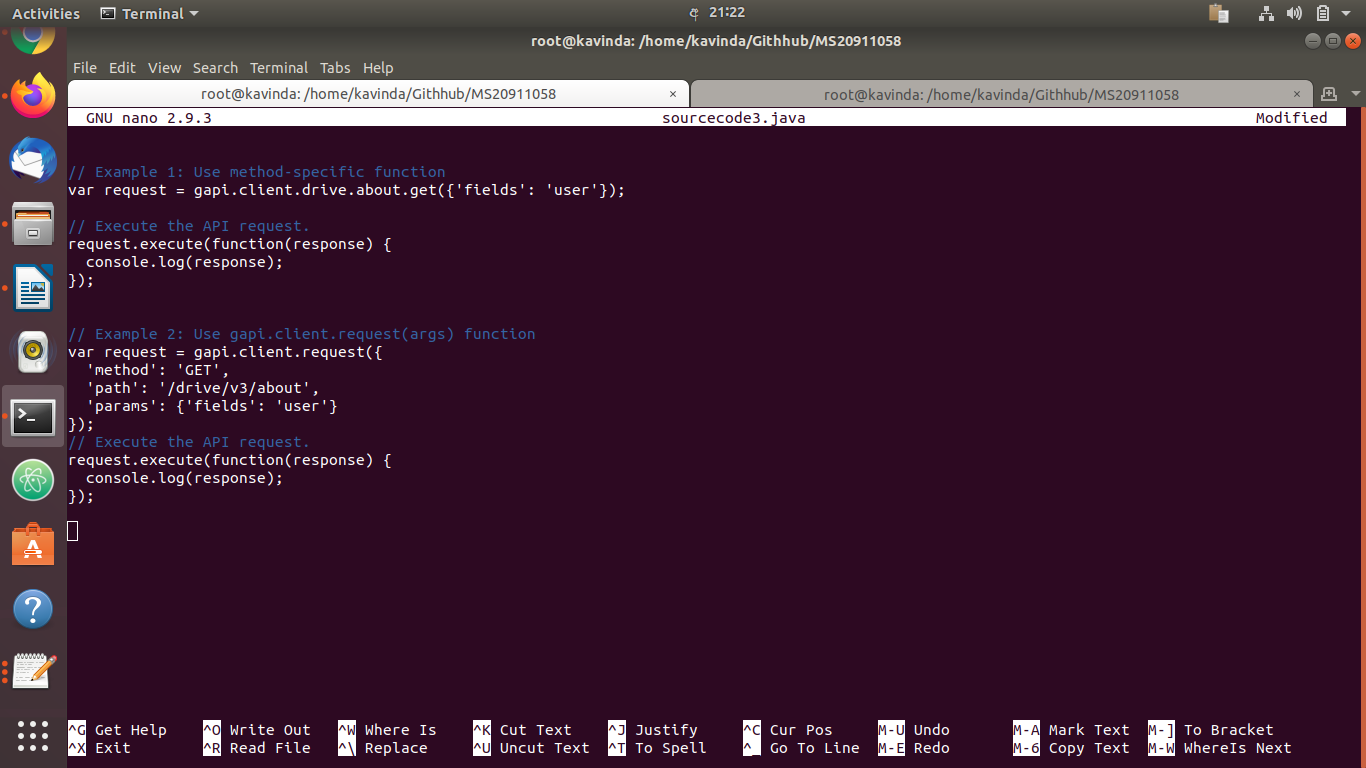
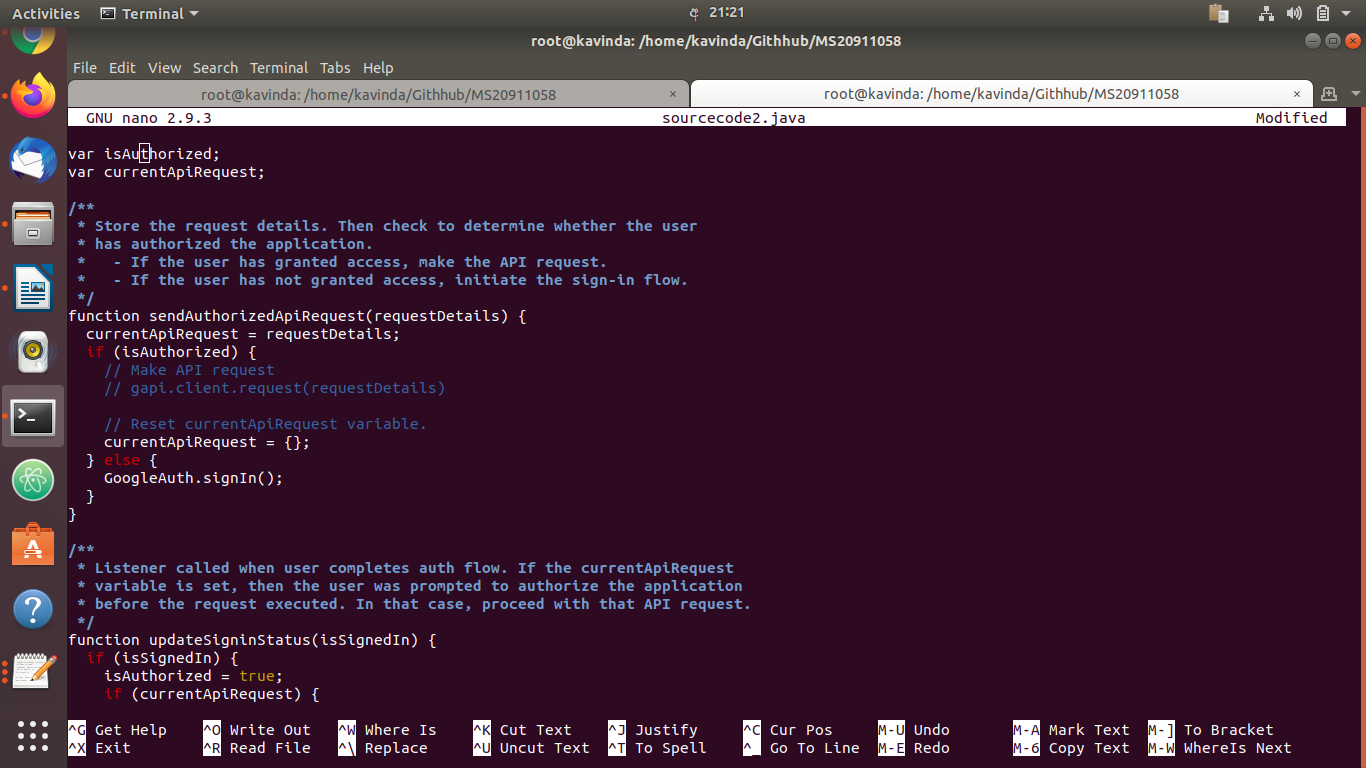


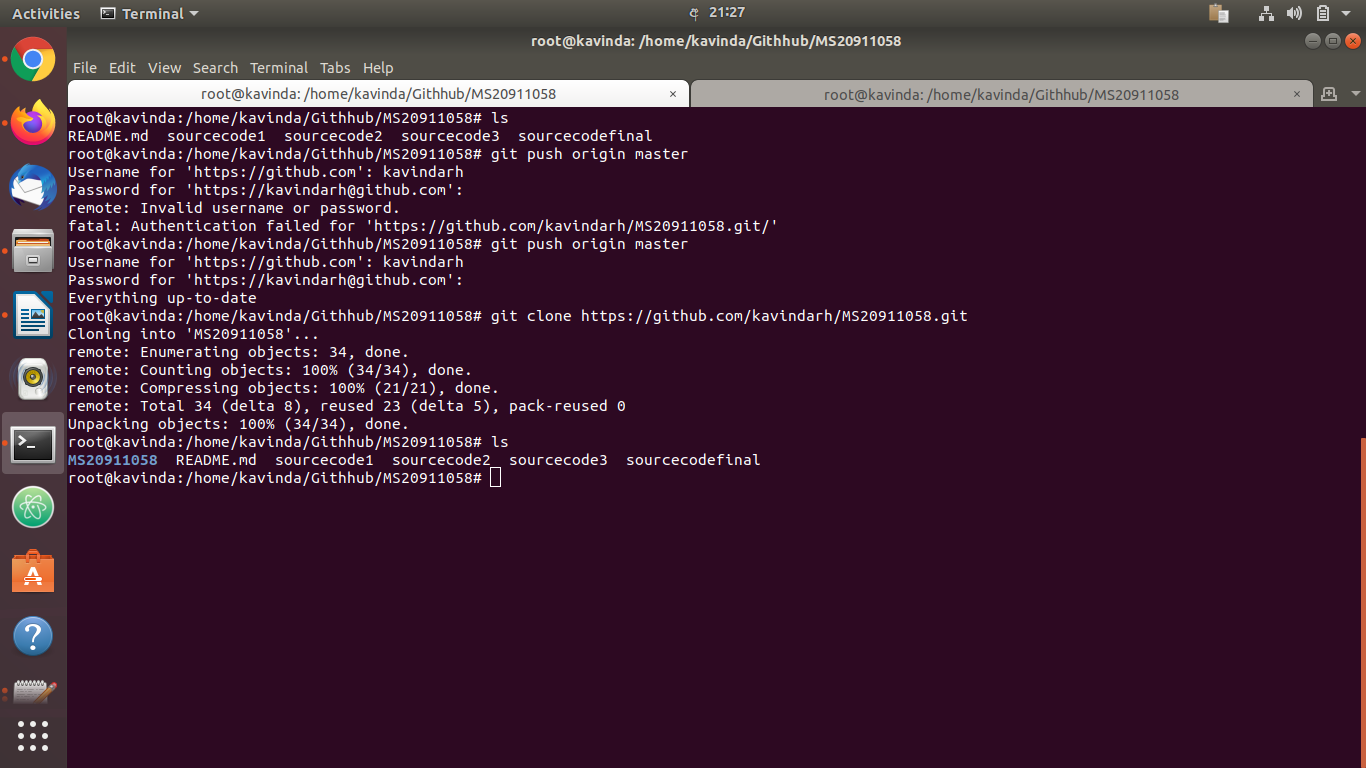
Figure 10

### HANDLE THE OAUTH 2.0 SERVER RESPONSE

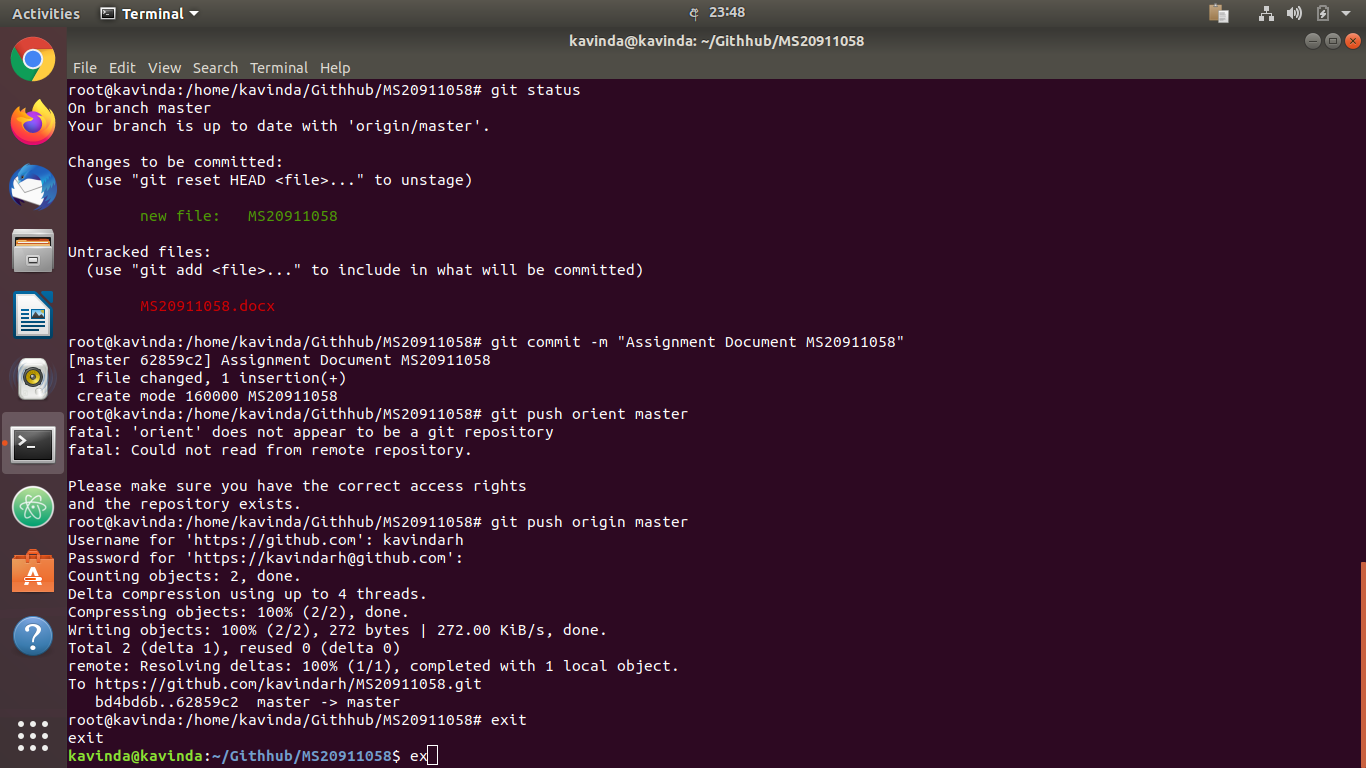
****

**Figure 11**

1. **PUSH TO GITHHUB**

****

**Figure 12**

****

**Figure 13**

**VII. CONCLUSION**

Except of traditional or legacy encryption methodologies, OAuth 2.0 would be a standard protection mechanism for critical data and information in this era. Oauth 2.0 protocol is most important services to be gained the protection between client and the third party applications. The token is the final authorization to precede the client request regardless of any other authentication. The key role is handled the token than the client password.

**REFERENCES**

# [1] D. Hard, Internet Engineering Task Force (IETF) “ The OAuth 2.0 Authorization Framework” RFC 6749, Microsoft, Category Standards Track ISSN: 2070-1721, October 2012.

[2] OKTA, <https://www.oauth.com/oauth2-servers/redirect-uris/>, My 2020

[3] Prabath Siriwardana and Johan Nellathammbi “OAuth 2.0", WSO2, [prabath@oso2.com](mailto:prabath@oso2.com), [john@oso2.com](mailto:john@oso2.com), [Online] Available: <https://slideplayer.com/slide/3923391/>, May 2020