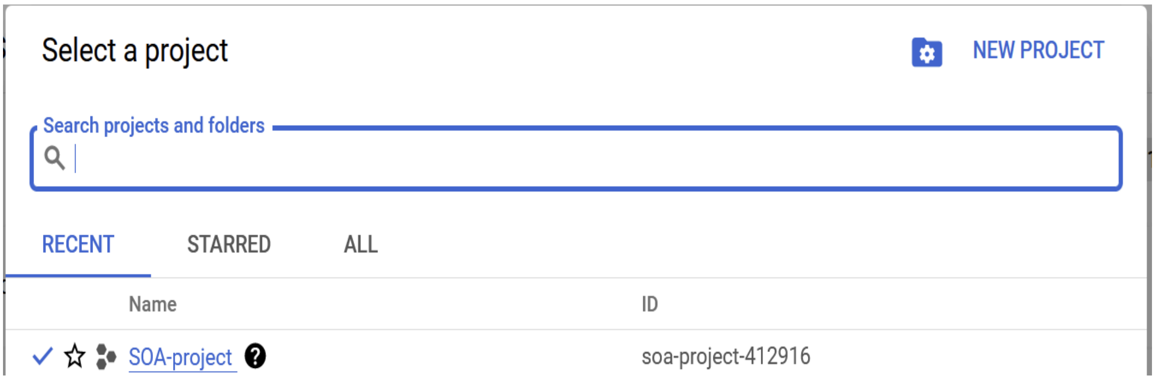
Secure APIs used Google Gmail API

A secure REST API refers to a Representational State Transfer (REST) API that has implemented robust security measures to protect sensitive data, prevent unauthorized access, and ensure the integrity of communication between clients and the server.

Google has several APIs that allow developers to use various Google resources in their applications. One of them is Gmail APIs that allows sending emails from your application to any address.

To create a secure REST API that interacts with the Google Gmail API using OAuth 2.0 in a Node.js application, you'll need to follow a series of steps. Below is a simplified guide to get you started:

1. **Create a Project in Google Cloud Console:**
    a. Go to the Google Cloud Console.
    b. Create a new project or select an existing one.
    c. Enable the Gmail API for your project.



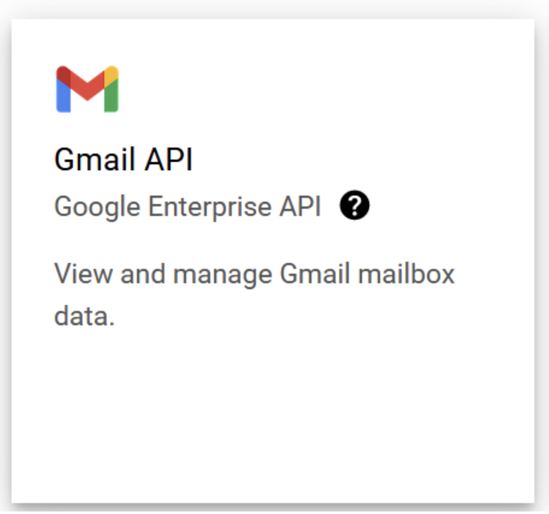
Select a project

NEW PROJECT

Search projects and folders

RECENT STARRED ALL

Name	ID
✓ ☆ SOA-project	soa-project-412916

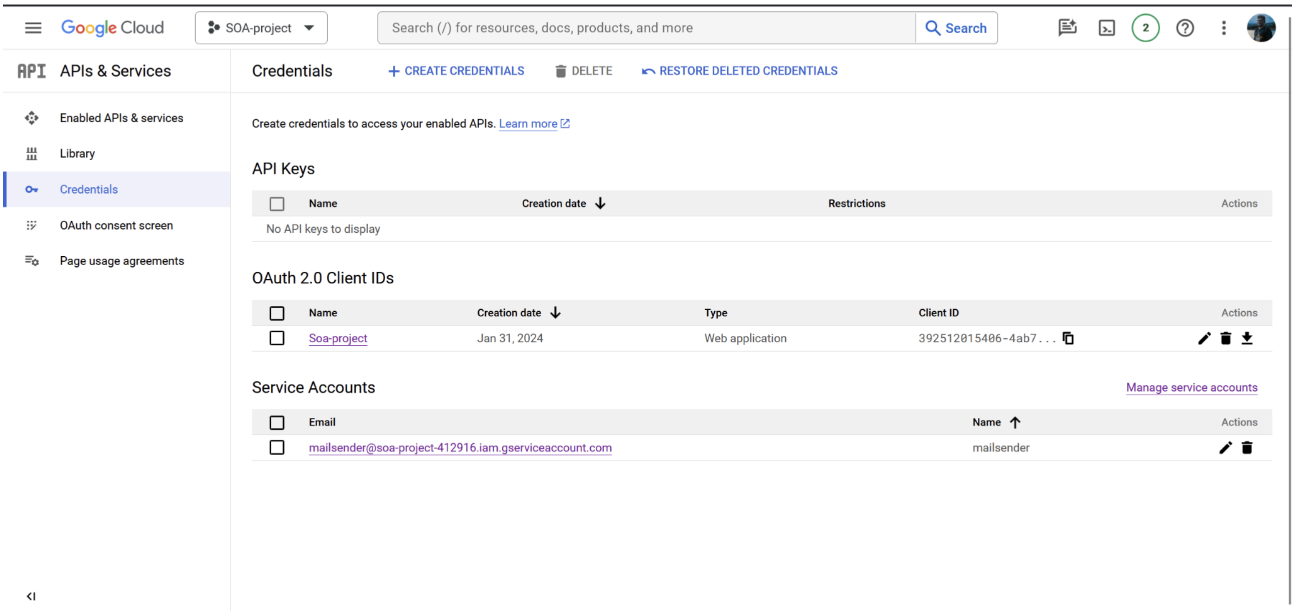


Gmail API

Google Enterprise API

View and manage Gmail mailbox data.

2. **Create OAuth 2.0 Credentials:**
    a. In the Cloud Console, navigate to "APIs & Services" > "Credentials."
    b. Create credentials and select "OAuth client ID."
    c. Configure the consent screen and set the authorized redirect URI.
    d. Download the JSON file containing your client ID and client secret.



Create credentials to access your enabled APIs. Learn more

API Keys

□	Name	Creation date ↓	Restrictions	Actions
No API keys to display				

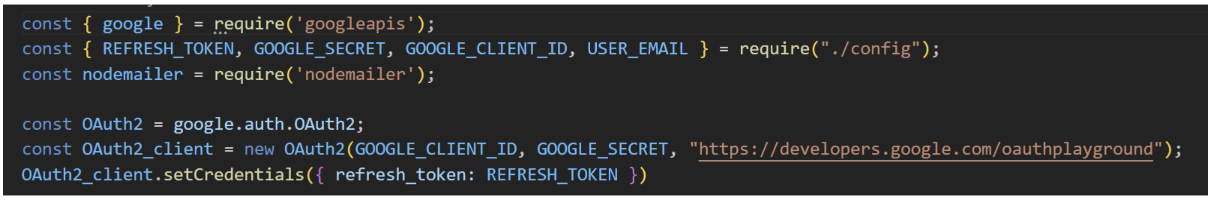
OAuth 2.0 Client IDs

□	Name	Creation date ↓	Type	Client ID	Actions
□	Soa-project	Jan 31, 2024	Web application	392512015406-4ab7...	

Service Accounts

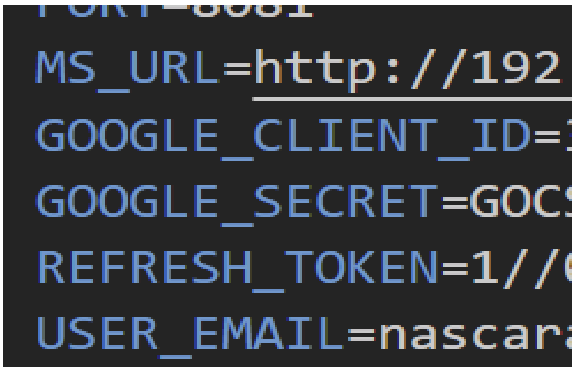
□	Email	Name ↑	Actions
□	mailsender@soa-project-412916.iam.gserviceaccount.com	mailsender	

3. **Set Up Your Node.js Application:**
    a. Install necessary packages with this command: **npm install express google-auth-library nodemailer**
    b. Implement OAuth2 Authentication



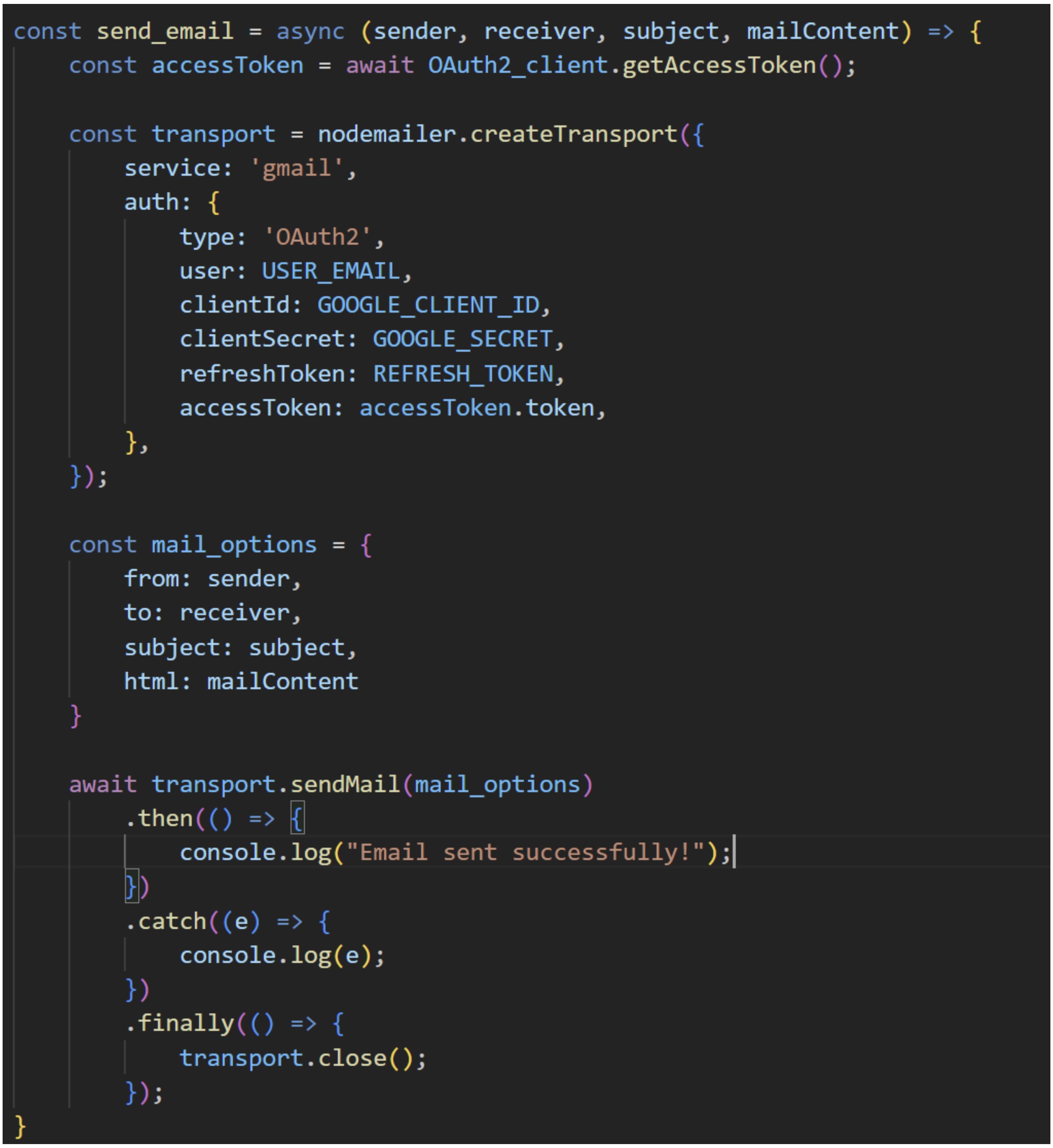
```
const { google } = require('googleapis');
const { REFRESH_TOKEN, GOOGLE_SECRET, GOOGLE_CLIENT_ID, USER_EMAIL } = require("./config");
const nodemailer = require('nodemailer');

const OAuth2 = google.auth.OAuth2;
const OAuth2_client = new OAuth2(GOOGLE_CLIENT_ID, GOOGLE_SECRET, "https://developers.google.com/oauthplayground");
OAuth2_client.setCredentials({ refresh_token: REFRESH_TOKEN })
```



```
PORT=8081
MS_URL=http://192
GOOGLE_CLIENT_ID=1
GOOGLE_SECRET=GOCS
REFRESH_TOKEN=1//0
USER_EMAIL=nascar
```

    c. Create an .env file for your variables
    d. Configure an API for sending emails



```
const send_email = async (sender, receiver, subject, mailContent) => {
    const accessToken = await OAuth2_client.getAccessToken();

    const transport = nodemailer.createTransport({
        service: 'gmail',
        auth: {
            type: 'OAuth2',
            user: USER_EMAIL,
            clientId: GOOGLE_CLIENT_ID,
            clientSecret: GOOGLE_SECRET,
            refreshToken: REFRESH_TOKEN,
            accessToken: accessToken.token,
        },
    });

    const mail_options = {
        from: sender,
        to: receiver,
        subject: subject,
        html: mailContent
    }

    await transport.sendMail(mail_options)
        .then(() => {
            console.log("Email sent successfully!");
        })
        .catch((e) => {
            console.log(e);
        })
        .finally(() => {
            transport.close();
        });
}
```