# Configurazione di google per usare le API di accesso a google calendar

Per poter configurare google al fine di accedere con i pacchetti nuget a google calendar occorre:

* creare un nuovo progetto in google (andando su <https://console.cloud.google.com>)
* abilitare le API necessarie
* creare le credenziali oauth client ID
* scaricare il file json e copiarlo in bin

## Accedi alla console di Google Cloud

* Vai su https://console.cloud.google.com
* Accedi con il tuo **account Google**.
* Se è la prima volta, Google ti chiederà di accettare i **termini di servizio** e di attivare la fatturazione (puoi sfruttare il **credito gratuito iniziale**).

## Crea un nuovo progetto

1. In alto, vicino al logo di Google Cloud, clicca sul **menu a tendina del progetto** (potrebbe dire "Seleziona progetto").
2. Clicca **"Nuovo progetto"**.
3. Compila i campi:
   * **Nome progetto** → può essere qualsiasi nome (es. calendar-integration).
   * **ID progetto** → generato automaticamente, puoi lasciarlo così.
   * **Organizzazione** → se non hai un’azienda collegata, lascia predefinito.
4. Clicca **"Crea"**.

## Abilita le API necessarie

Per usare Google Calendar API:

1. Con il progetto selezionato, vai su **"API e servizi" → "Libreria"**.
2. Cerca **"Google Calendar API"**.
3. Clicca **"Abilita"**.

## Crea le credenziali

1. Sempre in **"API e servizi"**, vai su **"Credenziali"**.
2. Clicca **"Crea credenziali"** → scegli **OAuth client ID** (per applicazioni utente) o **Service account** (per automazioni server).
3. Se scegli **OAuth client ID**:
   * Ti chiederà di **configurare la schermata di consenso**:
     + Inserisci nome app, email di supporto e (facoltativi) loghi e URL.
     + Salva.
   * Poi scegli **Tipo di applicazione**:
     + **Desktop App** (per test in locale con .NET Core)
     + **Web application** (se sarà un'app online)
   * Salva → scarica il file credentials.json.

## Perché appare la schermata di login??

1. Stai usando un **OAuth client ID di tipo “Installed”**.
2. La prima volta che esegui il codice, Google deve sapere **a quale account dare accesso al calendario**.
3. Ti mostra la schermata di login e poi chiede di autorizzare l’app (il nome “Calendar-integration” appare perché è il nome della tua app nelle impostazioni OAuth).
4. Dopo aver effettuato l’accesso e autorizzato, viene creato un file token.json che salva il token di accesso.
5. Alle esecuzioni successive, **non ti chiederà più l’account** finché token.json è valido.

## Accesso bloccato: l'app non ha completato la procedura di verifica di Google

succede perché:

Google permette l’uso in test **solo agli account email che hai aggiunto come tester** nella **schermata di consenso OAuth**.

Come risolvere (modalità sviluppo/test)

Vai nella console Google Cloud:

https://console.cloud.google.com/apis/credentials

Nel menu a sinistra, vai su Schermata di consenso OAuth.

Scorri fino alla sezione Utenti di test.

Clicca + AGGIUNGI UTENTI e inserisci l’email del Google account con cui vuoi fare login (es. licksandpatterns@gmail.com).

Salva.

## Alternative se vuoi evitare il fatto che ti chieda manualmente le credenziali dalla finestra del browser

Se vuoi **un’automazione server-side** senza login manuale:

1. Devi usare un **Service Account** invece di “Installed App”.
2. Con un Service Account:
   * Non serve aprire il browser.
   * Puoi dare accesso a calendari specifici condividendoli con l’email del Service Account.
   * Il codice usa il JSON del Service Account per autenticarsi.

### Creare un Service Account su Google Cloud

1. Vai su **Google Cloud Console – Service Accounts**.
2. Assicurati di avere selezionato **il progetto corretto** (quello dove hai abilitato Google Calendar API).
3. Clicca **Crea account di servizio**:
   * Nome: calendar-service
   * ID: generato automaticamente.
   * Descrizione: "Service account per accesso a Google Calendar".
4. Premi **Crea e continua**.
5. Nella parte “Concedi accesso al progetto” puoi **saltare** (non serve assegnare ruoli qui per Calendar).
6. Premi **Fine**.

### Scaricare la chiave JSON

1. Nella lista dei Service Account, clicca quello appena creato.
2. Vai nella scheda Chiavi.
3. Clicca Aggiungi chiave → Crea nuova chiave.
4. Scegli JSON e scarica il file.
5. Salvalo nella tua Console App (.NET Core), ad esempio service-account.json.
6. Non pubblicarlo mai su GitHub o repository pubblici.
7. Condividere il calendario con il Service Account
8. Il Service Account è un “utente virtuale” con una sua email, tipo:

### Per fare accedere l’utente virtuale a un calendario:

1. Vai su Google Calendar con il tuo account normale.
2. A sinistra, trova il calendario da condividere.
3. Clicca sui tre puntini → **Impostazioni e condivisione**.
4. Nella sezione “Condividi con persone specifiche” aggiungi **l’email del Service Account**.
5. Concedi il permesso **“Apportare modifiche e gestire la condivisione”**.

### Modifica dell API dot net

Quindi nel codice della api bisogna dare l’id del calendario e non semplicemente “primary”:

in google l’id del calendario lo trovi qui nel calendario che hai condiviso con l’utente fittizio che utilizza le tue API:

Immagine che contiene schermata, software, Software multimediale, testo

Il contenuto generato dall'IA potrebbe non essere corretto.

Questo è l’id da recuperare

Immagine che contiene testo, schermata, software, Software multimediale

Il contenuto generato dall'IA potrebbe non essere corretto.

Immagine che contiene testo, schermata, software, Software multimediale

Il contenuto generato dall'IA potrebbe non essere corretto.

L’id del calendario va scritto qui nel codice al posto di primary:

Immagine che contiene testo, Carattere, schermata, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

**Salva e proteggi il file credenziali**

* Metti credentials.json nella cartella del tuo progetto .NET Core.
* **Non** committarlo mai su GitHub o repository pubblici.
* **Da terminale (CLI .NET Core / Powershell / CMD)**
* Esegui questi comandi nella cartella del progetto (dove c’è il file .csproj):
* bash
* CopiaModifica
* dotnet add package Google.Apis.Auth
* dotnet add package Google.Apis.Calendar.v3